CASIO®

SH1001099C ㊥ Printed in Japan

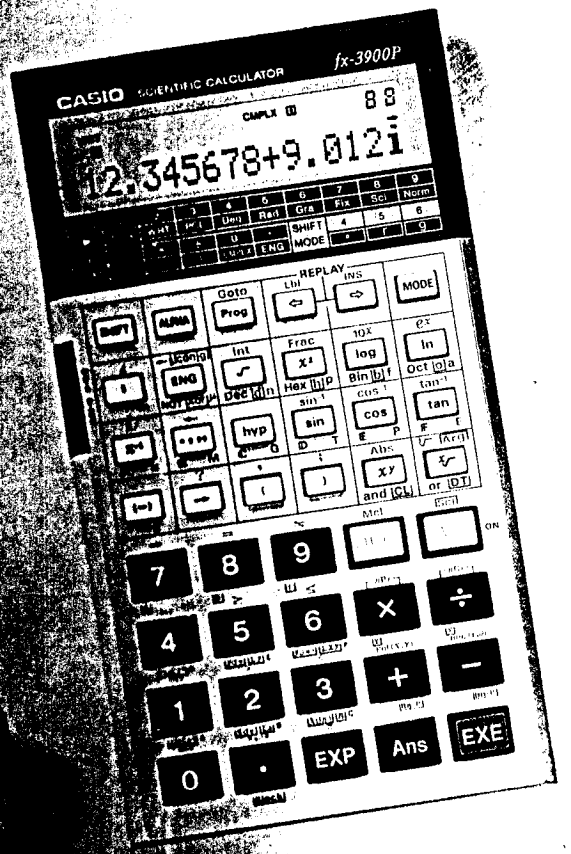
fx-3900P OWNER'S MANUAL

Master

fx-3900P

OWNER'S MANUAL

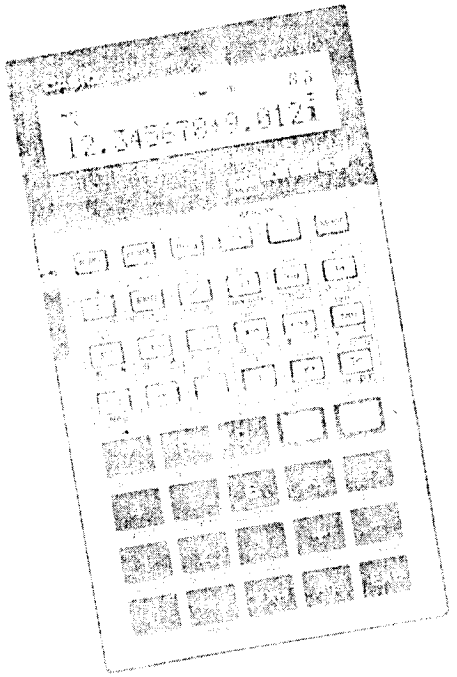
SPARE PARTS MASTER



CASIO®

fx-3900P

OWNER'S MANUAL



CASIO

- The information contained herein is subject to change without notice.
- Reproduction of this manual either in part or its entirety is forbidden.
- Note that the manufacturer assumes no responsibility for any injury or loss incurred while using this manual.
- Note that the manufacturer assumes no responsibility for any loss or claims by third parties which may arise through use of this unit.
- Note that the manufacturer assumes no responsibility for any damages incurred as a result of data loss caused by malfunctions, repairs or battery replacements. Physical records of important programs should be prepared to protect against such data losses.

FOREWORD

Thank you for your purchase of the CASIO fx-3900P.
This unit is an advanced hand held programmable computer capable of alphabetic display. Besides a programming function which is useful for performing repeat or complex computations, 101 scientific functions are also provided.
Manual computations can be easily performed following written formulas (true algebraic logic). A replay function is provided that allows confirmation or correction when key operation errors occur. Programs can also be input by following true algebraic logic, so repeat and/or complex computations are simplified.

This manual is composed of three sections:

1. Configuration and Operation
2. Manual Computations
3. Program Computations

Section 1 should be read first to become familiar with the nomenclature, handling and cautions concerning this unit. Sections 2 and 3 can then be read in order to master both types of computations through samples and explanations.

CONTENTS

FOREWORD i

HANDLING PRECAUTIONS vi

1. CONFIGURATION AND OPERATION 1

1-1 NOMENCLATURE AND FUNCTIONS 2
- Display window 3
- Power switch 3
- Special operation keys 4
- Numeric/Decimal point keys 6
- Computation keys 7
- Function keys 7
- Contrast adjustment dial 11

1-2 POWER AND BATTERY REPLACEMENT 12
- Procedure 12

1-3 BEFORE BEGINNING COMPUTATIONS 14
- Computation priority sequence 14
- Number of stacks 15
- Computation modes 16
- Number of input/output digits and computation digits 17
- Overflow and errors 17
- Number of input characters 18
- Corrections 19
- Memory 20
- Answer (Ans) function 21
- Auto power off function 22

2. MANUAL COMPUTATIONS 23

2-1 BASIC COMPUTATIONS 24
- Arithmetic operations 24
- Parenthesis computations 25
- Memory computations 26
- Specifying the number of decimal places, the number of significant digits and the exponent display 27

2-2 SPECIAL FUNCTIONS 29
- Continuous computation function 29
- Replay function 30
- Multistatement function 31

2-3 FUNCTIONAL COMPUTATIONS 32
- Angular measurement units 32
- Trigonometric functions and inverse trigonometric functions 33
- Logarithmic and exponential functions 34
- Hyperbolic functions and inverse hyperbolic functions 35
- Coordinate transformation 36
- Permutation and combination 37
- Other functions 38

2-4 ENGINEERING SYMBOL COMPUTATIONS 40

2-5 BINARY, OCTAL, DECIMAL, HEXADECIMAL COMPUTATIONS 41
- Binary, octal, decimal, hexadecimal conversions 42
- Negative expressions 43
- Basic arithmetic operations using binary, octal, decimal and hexadecimal values 43
- Logical operations 44

2-6 COMPLEX NUMBER COMPUTATIONS 45
- General computations 45
- Other computations 46

2-7 STATISTICAL COMPUTATIONS 47
- Standard deviation 47
- Regression computation 49
  - Linear regression 50
  - Logarithmic regression 51
  - Exponential regression 52
  - Power regression 53

3. PROGRAM COMPUTATIONS 55

3-1 WHAT IS A PROGRAM? 56
- Formulas 56
- Programming 56
- Program storage 57
- Program execution 59

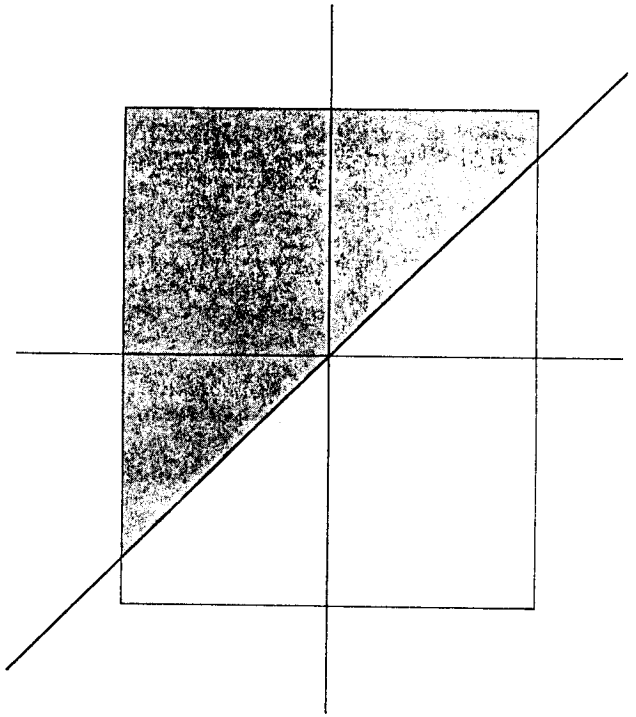
3-2 PROGRAM CHECKING AND EDITING (CORRECTION, ADDITION, DELETION) 60
Formulas 60
Programming 60
Program editing 61
Program execution 62
Summary 62
3-3 PROGRAM DEBUGGING (CORRECTING ERRORS) 63
Debugging when an error message is generated 63
Error messages 64
Checkpoints for each type of error 64
3-4 COUNTING THE NUMBER OF STEPS 66
3-5 PRECAUTIONS WHEN USING FUNCTIONS 67
3-6 PROGRAM AREAS AND COMPUTATION MODES 68
Program area and computation mode specification in the WRT mode 68
Cautions concerning the computation modes 69
3-7 ERASING PROGRAMS 71
Erasing a single program 71
Erasing all programs 71
3-8 CONVENIENT PROGRAM COMMANDS 72
Jump commands 72
Unconditional jump 72
Conditional jump 74
Count jump 76
Summary 78

PROGRAM LIBRARY 79
Greatest common measure 80
Definite integrals using the Simpson's rule 82
Minimum loss matching 84
Cantilever under concentrated load 86
Parabolic movement 88
REFERENCE MATERIAL 97
Manual computations 98
Program computations 102
Error messages 104
Input range of functions (general principles) 105
SPECIFICATIONS 107

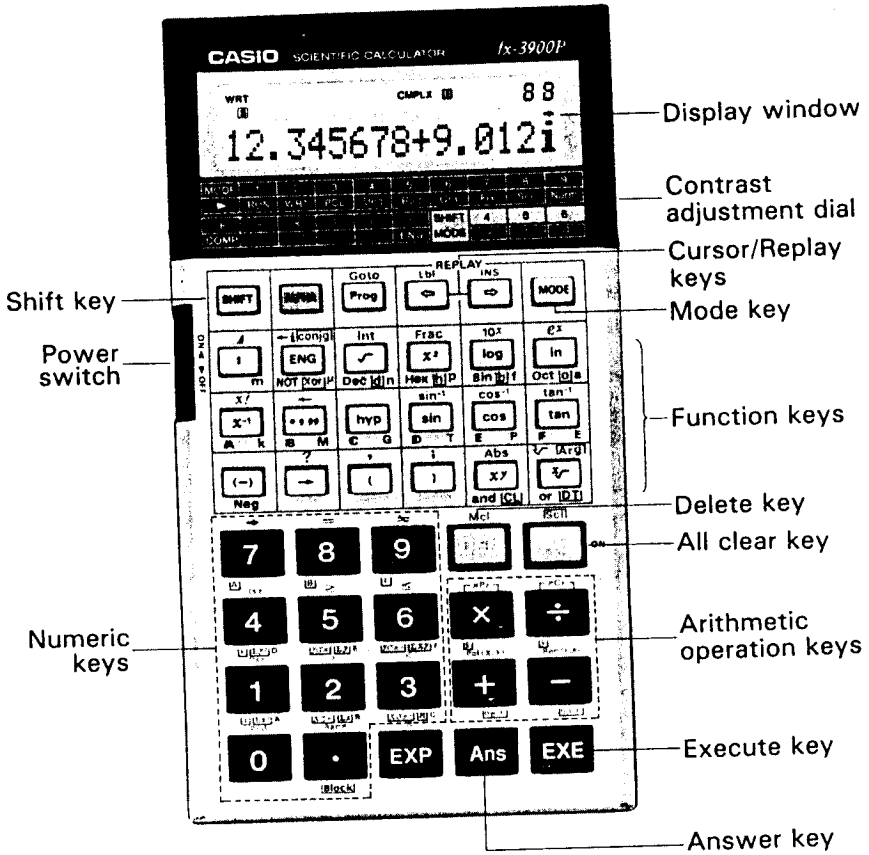
HANDLING PRECAUTIONS

- This unit is composed of precision electronic components, and should never be disassembled. Do not drop it or otherwise subject it to sudden impacts, or sudden temperature changes. Be especially careful to avoid storing the unit or leaving it in areas exposed to high temperature, humidity or large amounts of dust. When exposed to low temperatures, the unit will require more time to display answers and may even fail to operate. The display will return to normal once normal temperature is attained.
- Batteries should be replaced every 2 years even if the unit is not used for extended periods. Never leave dead batteries in the battery compartment. They can leak and cause damage to the unit.
- Avoid using volatile liquids such as thinner or benzine to clean the unit. Wipe the unit with a soft, dry cloth or a cloth that has been dipped in a neutral detergent solution and wrung out.
- If malfunction of the unit should occur, either bring or send the unit to your retailer or the nearest CASIO dealer.
  Be sure to clearly explain the problem in detail.
- Before assuming malfunction of the unit, be sure to carefully reread this manual and ensure that the problem is not due to insufficient battery power, programming or operational errors.

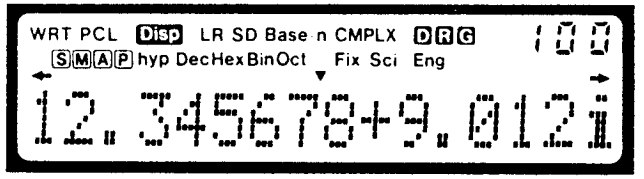
1.
CONFIGURATION AND OPERATION



1-1 NOMENCLATURE AND FUNCTIONS



■ Display window



A "status display"is located at the top of the display window to provide information on the status of the computer. The status display illuminates for each of the operational modes (see page 4).
D, R, G are units of angular measurement and stand for degrees, radians and grads respectively.
S illuminates when the SHIFT key is pressed.
M illuminates when the MODE key is pressed.
A illuminates when the ALPHA key is pressed to put the unit in the alphabet mode.
P illuminates when the Prog key is pressed.
"Fix"and"Sci" illuminate when specifying the number of decimal places or the number of significant digits.
The three-digit number located in the upper right portion of the display window indicates the number of remaining steps.
Formulas and/or computation results are shown at the bottom of the display. Zeros are indicated as 0 to distinguish them from the letter O.
Subsequent inputs of letters or numbers will begin from the point indicated by the cursor.
A maximum of twelve characters can be shown on the display window at one time. The ← and → indicate that the character string being displayed exceeds the capacity of the display window in the respective direction indicated.

■ Power switch

Power is turned ON by sliding the power switch up. Sliding the power switch down turns power OFF.

■ Special operation keys

[SHIFT] Shift key

Press when using the function commands and functions marked in brown on the key panel. An [S] will illuminate on the display to indicate that [SHIFT] has been pressed. Pressing [SHIFT] again will cause the [S] to disappear from the display and the unit to return to the status it was in before [SHIFT] was originally pressed.

[MODE] Mode key

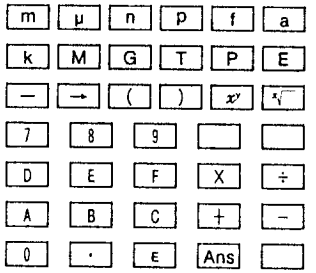
Press when setting the status of the unit or the unit of angular measurement.

- [MODE] [0] ··· CMPLX illuminates on display. For complex number computations.
- [MODE] [1] ··· For manual computations and program execution.
- [MODE] [2] ··· WRT illuminates on display. For writing or checking programs.
- [MODE] [3] ··· PCL illuminates on display. For clearing programs.
- [MODE] [4] ··· Deg illuminates on display. If [EXE] is pressed, **D** illuminates on display and unit of angular measurement is specified as degrees.
- [MODE] [5] ··· Rad illuminates on display. If [EXE] is pressed, **R** illuminates on display and unit of angular measurement is specified as radians.
- [MODE] [6] ··· Gra illuminates on display. If [EXE] is pressed, **G** illuminates on display and unit of angular measurement is specified as grads.
- [MODE] [7] ··· Fix displayed. Entering a value from 0 to 9 followed by [EXE] will specify the number of decimal places according to the value entered.
  Ex. [MODE] [7] [3] [EXE] → Three decimal places
- [MODE] [8] ··· Sci displayed. Entering a value from 0 to 9 followed by [EXE] will specify the number of significant digits from 1 to 10.
  Ex. [MODE] [8] [5] [EXE] → 5 significant digits
- [MODE] [9] ··· Norm displayed. Pressing [EXE] will cancel the specified number of decimal places or the specified number of significant digits.
- [MODE] [·] ··· Eng displayed. If [EXE] is pressed, ENG illuminates on display to indicate engineering symbol computation mode (engineering floating point).

- [MODE] [+] ··· Specifies COMP mode for arithmetic computation or function computation (program execution possible).
- [MODE] [−] ··· Base-n illuminates. For binary, octal or hexadecimal computations/conversions.
- [MODE] [×] ··· SD illuminates. For standard deviation computations.
- [MODE] [÷] ··· LR illuminates. For regression computations.
- [SHIFT] [MODE] [4] ··· Pressed after a numeric value representing degrees is input.
- [SHIFT] [MODE] [5] ··· Pressed after a numeric value representing radians is input.
- [SHIFT] [MODE] [6] ··· Pressed after a numeric value representing grads is input.

[ALPHA] Alphabet key

Press to input alphabetic characters or engineering symbols. Pressing [ALPHA] displays [A] and allows the input of only one character. After that, the unit returns to the status it was in before the [ALPHA] key was originally pressed.



[Goto Prog] Program/Goto key

Press [Prog] and enter a value from 1 to 4 to execute a program.

Ex. [Prog] [1] → Execution of Program 1 begins.

Pressing [SHIFT] followed by [Goto] ([Goto Prog] key) will cause Goto to appear on the display. This is a jump command used in programs.

[LBL REPLAY INS] [⇦] [⇨] Cursor/Replay/Label/Insert keys

Press to move the cursor to the left or right on the display to correct formulas or numeric values.

Pressing [⇦] moves the cursor to the left, while pressing [⇨] moves the cursor to the right. Pressing either key and holding it down will cause continuous movement of the cursor in the respective direction. Once a formula or numeric value is input and [EXE] is pressed,

these keys become replay keys. Pressing ⇦ displays the formula or numeric value from the end, and pressing ⇨ displays it from the beginning. Pressing EXE again will re-execute.
Re-execution can also be performed after changes have been made in the formula or numeric value. (See page 30.)
Pressing SHIFT followed by Lbl (Lbl ⇦ key) will input a label into the program, while pressing SHIFT followed by INS (INS ⇨ key) will specify the insert mode.

DEL Delete key
Press to delete the character at the current position of the cursor. When the character is deleted, everything to the right of the cursor position will shift one space to the left.
Pressing SHIFT DEL EXE will clear the memory contents.

AC All clear key
Press to completely clear the formula or numeric value displayed. Also used to clear errors indicated by error message displays, and to restore power after activation of the auto power off function. (See page 22.)

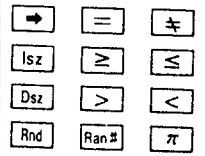
EXE Execute key
Press to obtain the result of a computation. Pressed after data input for a programmed computation or to advance to the next execution after a computation result is obtained.

Ans Answer key
Pressing Ans followed by EXE will recall the last computation result. It can be recalled by Ans EXE even after it has been cleared using the AC key or by switching the power of the unit OFF. When used during program execution, the last result computed is recalled.

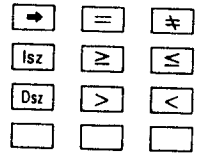
■ 1 ~ 9, · Numeric/Decimal point keys

When entering numeric values, enter the number in order. Press the · key to enter the decimal point in the desired position.
SHIFT key combinations for the various modes are as follows:

COMP mode (MODE +)
CMPLX mode (MODE 0)
ENG mode (MODE ·)

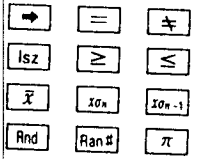


Base-n mode (MODE −)



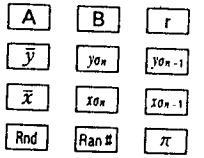
*n*P*r*, *n*C*r*, Pol(, Rec(, Rnd, Ran # and π cannot be used in this mode.

SD mode (MODE ×)



Standard deviation functions can be used.

LR mode (MODE ÷)



Paired variable statistic functions can be used.

■ Computation keys

+ − × ÷ Arithmetic operation keys
For addition, subtraction, multiplication and division, enter the computation as it reads. SHIFT key combinations for the various modes are as follows:

COMP mode or SD mode

nPr nCr (nPr × and nCr ÷ keys)···Permutation/combination
Pol(Rec((Pol(x,y) + and Rec(r,θ) − keys)···Coordinate transformation

LR mode

$\hat{x}$ $\hat{y}$ ($\hat{x}$ × $\hat{y}$ ÷ keys)···Estimated value computation of x and y
Pol(Rec(···Coordinate tarnsformation

- Pressing after SHIFT in the CMPLX allows + to be used for real number extraction, and − for imaginary number extraction.

■ Function keys

Press for functional computation. Various uses are available in combination with the SHIFT key, and/or depending on the mode being used.

[:] Multistatement/Display key

- Press to separate formulas or commands in programmed computations or consecutive computations.
  The result of such combinations is known as a multistatement.
  (See page 31.)
- When pressed following the [SHIFT] key, the results of each section of the programmed computations or consecutive computations are sequentially displayed with each press of [EXE].

[ENG] Engineering/Negation key

- Press to convert a computation result to an exponential display whose exponent is a multiple of three. (10^3=K, 10^6=M, 10^9=G, 10^{-3}=m, 10^{-6}=μ, 10^{-9}=n, 10^{-12}=p)
- When obtaining logical negation for a value in the Base–n mode, press prior to entering the value.
- Press folowing the [SHIFT] key in the Base-n mode to obtain an exclusive OR (Xor).
- Press in the CMPLX mode to enter "i".
- Press following the [SHIFT] key in the CMPLX mode to obtain the conjugate complex number.

[√] Root/Integer key

- Press prior to entering a numeric value to obtain the square root of that value.
- When pressed following the [SHIFT] key, the integer portion of a value can be obtained.
- Press followed by [EXE] in the Base-n mode to specify the decimal computation mode.
- When pressed following the [SHIFT] key in the Base-n mode, the subsequently entered value is specified as a decimal value.

[x^2] Square/Fraction key

- Press after a numeric value is entered to obtain the square of that value.
- When pressed following the [SHIFT] key, the decimal portion of a value can be obtained.
- Press followed by [EXE] in the Base-n mode to specify the hexadecimal computation mode.
- When pressed following the [SHIFT] key in the Base-n mode, the subsequently entered value is specified as a hexadecimal value.

[log] Common logarithm/Antilogarithm key

- Press prior to entering a value to obtain the common logarithm of that value.
- When pressed following the [SHIFT] key, the subsequently entered value becomes an exponent of 10.
- Press followed by [EXE] in the Base-n mode to specify the binary computation mode.
- When pressed following the [SHIFT] key in the Base-n mode, the subsequently entered value is specified as a binary value.

[ln] Natural logarithm/Anti-natural logarithm key

- Press prior to entering a value to obtain the natural logarithm of that value.
- When pressed following the [SHIFT] key, the subsequently entered value becomes an exponent of e.
- Press followed by [EXE] in the Base-n mode to specify the octal computation mode.
- When pressed following the [SHIFT] key in the Base-n mode, the subsequently entered value is specified as an octal value.

[x^{-1}] Reciprocal/Factorial key

- Press prior to entering a value to obtain the reciprocal of that value.
- When pressed following the [SHIFT] key, the factorial of a previously entered value can be obtained.
- Press in the Base-n mode to enter A (10_{10}) of a hexadecimal value.

[°' "] Degree/minute/second key (decimal ↔sexagesimal key)

- Press to enter sexagesimal value.
  (degree/minute/second or hour/minute/second)
  Ex. 78°45' 12"→78 [°' "] 45 [°' "] 12 [°' "]
- When pressed following the [SHIFT] key, a decimal based value can be displayed in degrees/minutes/seconds.
  (hours/minutes/seconds).
- Press in the Base-n mode to enter B (11_{10}) of a hexadecimal value.

[hyp] Hyperbolic key

- Pressing [hyp], and then [sin], [cos], or [tan] prior to entering a value produces the respective hyperbolic function (sinh,cosh,tanh) for the value.

- Pressing [SHIFT], then [hyp] and then [sin], [cos], or [tan] prior to entering a value produces the respective inverse hyperbolic function ($\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$) for the value.
- Press in the Base-n mode to enter C (12_{10}) of a hexadecimal value.

[sin] [cos] [tan] Trigonometric function/Inverse trigonometric function keys

- Press one of these keys prior to entering a value to obtain the respective trigonometric function for the value.
- Press [SHIFT] and then one of these keys prior to entering a value to obtain the respective inverse trigonometric function for the value.
- Press in the Base-n mode to enter D, E, F(13_{10}, 14_{10}, 15_{10}) of a hexadecimal value.

[(–)] Minus key

- Press prior to entering a numeric value to make that value negative.
  Ex. $-123 \rightarrow$ [(–)] [1] [2] [3]
- Press in the Base-n mode prior to entering a value to obtain the negative of that value. The negative number is the twos complement of the value entered.

[→] Assignment key

- Press prior to entering a memory to assign the result of a computation to that memory.
  Ex. To assign the result of 12 + 45 to memory A: [1] [2] [+] [4] [5] [→] [ALPHA] [A] [EXE]
- During execution of program computations, press following the [SHIFT] key to enter a numeric value.

[(] [)] Parenthesis keys

- Press the open parenthesis key and the closed parenthesis key at the position required in a formula.
- When pressed following the [SHIFT] key, a comma or semicolon can be inserted to separate the arguments in coordinate transformation or consecutive computations.

[x^y] Power/Absolute value key

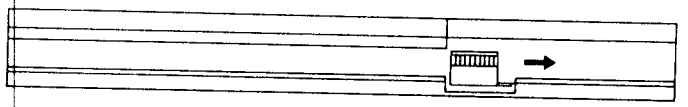
- Enter x (any number), press this key and then enter y (any number) to compute x to the power of y.
  In the SD or LR mode, this function is only available after pressing the [SHIFT] key.
- Press following the [SHIFT] key to obtain the absolute value of a subsequently entered numeric value.
- Press in the Base-n mode to obtain a logical product ("and").
- Press in the SD or LR mode to delete input data.

[$\sqrt{\ }$] [$\sqrt[x]{\ }$] Root/Cube root key

- Enter x, press this key and then enter y to compute the xth root of y. In the SD or LR mode, this function is only available after pressing the [SHIFT] key.
- Press following the [SHIFT] key to obtain the cube root of a subsequently entered numeric value.
- Press in the Base-n mode to obtain a logical sum ("or").
- Used as a data input key in the SD or LR mode.
- Pressing after [SHIFT] in the CMPLX mode produces the argument in complex number computations.

■ Contrast adjustment dial

From different viewing angles or due to weakening batteries, the display may sometimes appear too dark or too light. If this should occur, adjust the contrast of the display using the dial located on the right edge of the unit.



Moving the dial in the direction indicated by the arrow will cause the characters on the display to become stronger, while moving it in the opposite direction will cause them to become weaker. If, after maximum adjustment, the characters on the display still are too weak, it can indicate that battery power is too low.
In this case the batteries should be replaced as soon as possible.

1-2 POWER AND BATTERY REPLACEMENT

Power is supplied to this unit by two lithium batteries (CR2032). If the power of the batteries should diminish, the display will weaken and become difficult to read. A weak display even after contrast adjustment (see page 11) may indicate power is too low, so the batteries should be replaced. When making replacements, be sure to replace both batteries.

** If batteries are used for longer than two years, there is the danger of leakage. Be sure to replace batteries at least once every two years even if the unit is not used during that period.*

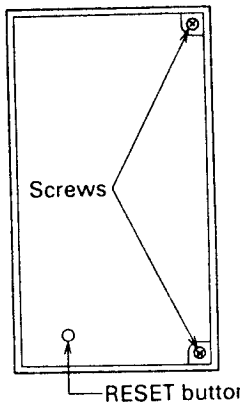
** Stored programs or data are erased when batteries are replaced. Therefore, it is recommended that programs and data required for later use be recorded on a coding sheet before replacing batteries.*

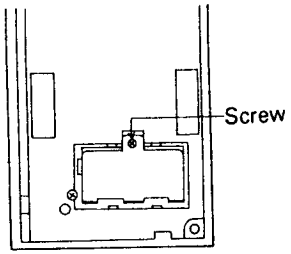
■ Procedure

① Slide the power switch to the OFF position, remove the two screws on the back of the unit with a screwdriver, and remove the back cover.

② Remove the screw holding the battery pressure plate in place and then remove the battery pressure plate.

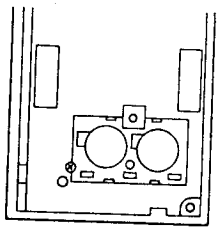
③ Remove both of the old batteries from the unit.
(This can be done easily by turning the unit so the battery compartment is facing downwards, and then lightly tapping the unit.)





④ Wipe the surfaces of two new batteries with a soft, dry cloth and load them into the unit ensuring that the positive ⊕ sides are facing upwards.

⑤ Fasten the battery pressure plate in place using the screw, and replace the back cover.



** IMPORTANT: Never dispose of old batteries in such a way that they will be incinerated. Batteries may explode if exposed to fire.*

CAUTIONS:

If the batteries being replaced are not totally without power, it is possible to replace batteries so quickly that previously stored programs and memory contents are not erased or altered. In this case, however, all programs and memory contents should be carefully checked after battery replacement.

If battery power should be allowed to decrease or if batteries are removed from the unit for extended periods, programs and memory contents may be erased or altered. In this case, the RESET button located on the back of the unit should be pressed using pointed object.

All memory contents and programs will be erased.

Keep batteries out of the reach of small children. If a battery should inadvertantly be swallowed, contact a doctor immediately.

1-3 BEFORE BEGINNING COMPUTATIONS.....

■ Computation priority sequence

This unit employs true algebraic logic to compute the parts of a formula in the following order:

1. Coordinate transformation Pol(x,y), Rec(r,θ)
2. Type A functions★ x^2, x^{-1}, $x!$, °, ′, ″, °′″
3. Power/root x^y, $\sqrt[x]{\ }$
4. Abbreviated multiplication format in front of π, memory, or parenthesis 2π, 4A 3(5+6), etc.
5. Type B functions★ $\sqrt{\ }$, $\sqrt[3]{\ }$, log, 10^x, ln, e^x, sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, (−), Abs, Int, Frac, h, d, b, o, Neg, Not
6. Abbreviated multiplication format in front of Type B functions 3sin5, $6\sqrt{7}$, 2sin30cos60, etc.
7. Permutation, combination nPr, nCr
8. ×,÷
9. +,−
10. and
11. or, xor
12. Relational operators <, >, =, ≠, ≤, ≥

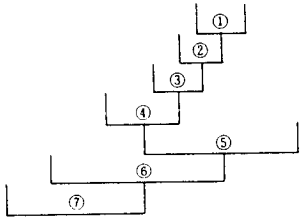
★ Functions are divided into two types.
Type A functions are entered after the argument, while Type B functions are entered before the argument.

** When functions with the same priority are used in series, execution is performed from right to left : e.g., $e^x\ln\sqrt{120} \rightarrow e^x\{\ln(\sqrt{120})\}$. Otherwise, execution is from left to right.*

** Compound functions are executed from right to left :*
e.g., $\sin\cos^{-1}0.6 \rightarrow \sin(\cos^{-1}0.6)$.

** Everything contained within parentheses receives highest priority.*

Ex. $2+3\times(\log\sin 2\pi^2\,\text{rad}+6.8)=22.07101691$

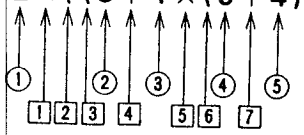


■ Number of stacks

This unit features a memory known as a stack for the temporary storage of low priority numeric values and commands (functions, etc). The numeric value stack has six levels, while the command stack has fourteen. If a complex formula is employed that exceeds the stack space available, a stack error (Stk ERROR) message will appear on the display.

Ex. Stack counting method

$2\times(3+4\times(5+4)\div 3)\div 5)+8=$



Numeric value stack	
①	2
②	3
③	4
④	5
⑤	4
⋮	

Command stack	
1	×
2	(
3	(
4	+
5	×
6	(
7	+
⋮	

** Computations are performed in the order of the highest computation priority first. Once a computation is executed, it is cleared from the stack.*

■ Computation modes

This unit features modes for manual computations, storing programs, and modes for general as well as statistical computations. The proper mode to suit computational requirements should be employed.

● Operation modes

There are a total of three operation modes.

1. RUN mode
   Manual computations including functional computation and program execution.
2. WRT mode
   Program storage and editing. (See Section 2.) In this mode, WRT illuminates at the left edge of the display window.
3. PCL mode
   Deletion of stored programs. (See Section 2.) In this mode, PCL illuminates at the left edge of the display window.

● Computation modes

There are a total of six computation modes which are employed according to the type of computation.

1. ENG mode
   Used to perform ENG computations involving engineering units. ENG displayed while this mode is active.
2. CMPLX mode
   Used to perform complex number computations. CMPLX displayed while this mode is active.
3. COMP mode
   General computations, including functional computations.
4. Base-n mode
   Binary, octal, decimal, hexadecimal conversion and computations, as well as logical operations. (See page 41.) Function computations can not be performed. In this mode, Base-n illuminates at the center top of the display window.
5. SD mode
   Standard deviation computation (1-variable statistics). (See page 47.) In this mode, SD illuminates at the center top of the display window.
6. LR mode
   Regression computation (paired variable statistics). (See page 49.) In this mode, LR illuminates at the center top of the display.

With so many modes available, computations should always be performed after confirming which mode is active.

■ Number of input/output digits and computation digits

● The allowable input/output range (number of digits) of this unit is 10 digits for a mantissa and 2 digits for an exponent. Computations, however, are internally performed with a range of 12 digits for a mantissa and 2 digits for an exponent.

Ex. $3 \times 10^5 \div 7 =$

3 [EXP] 5 [÷] 7 [EXE]

3 [EXP] 5 [÷] 7 [−] 42857 [EXE]

42857.14286
0.1428571

** Computation results greater than 10^{10} (10 billion) or less than 10^{-1} (0.1) are automatically displayed in exponential form.*

Ex. $123456789 \times 9638 =$

123456789 [×] 9638 [EXE]

1.189876532_E 12

Mantissa Exponent

Once a computation is completed, the mantissa is rounded off to 10 digits and displayed. And the displayed mantissa can be used for the next computation.

Ex. $3 \times 10^5 \div 7 =$

3 [EXP] 5 [÷] 7 [EXE]

[÷] 42857 [EXE]

42857.14286
0.14286

■ Overflow and errors

If the computational range of the unit is exceeded, or incorrect inputs are made, an error message will appear on the display window and subsequent operation will be impossible. This is the error check function. The following operations will result in errors:

(1) The answer, whether intermediate or final, or any value in memory exceeds the value of $\pm 9.999999999 \times 10^{99}$.

(2) An attempt is made to perform functional computations that exceed the input range. (See page 105.)

(3) Improper operation during statistical computations.
(Ex. Attempting to obtain $\bar{x}$ or $x\sigma_n$ without data input.)

(4) The capacity of the numeric value stack or the command stack is exceeded.
(Ex. Entering thirteen successive [(]'s followed by [2] [+] [3] [×] [4])

(5) Input errors are made.
(Ex. [5] [+] [+] [3] [EXE])

The following error messages will be displayed for the operations noted above:

(1) ~ (3) Ma ERROR
(4) Stk ERROR
(5) Syn ERROR

Besides these, there is a "Go ERROR". This error mainly occurs when using programs. See page 64 or the Error Message Table on page 104.

■ Number of input characters

This unit features a 60-step area for computation execution.
One function comprises one step. Each press of numeric or [+], [−], [×], and [÷] keys comprise one step. Though such operations as [SHIFT] [x!] ([x^{-1}] key) require two key operations, they actually comprise only one function and, therefore, only one step.
These steps can be confirmed using the cursor. With each press of the [◁] or [▷] key the cursor is moved one step.
Input characters are limited to 60 steps. Usually the cursor is represented by a blinking "–", but once the 54rd step is reached the cursor changes to a blinking "■". If the"■" appears during a computation, the computation should be divided at some point and performed in two parts.

** When numeric values or computation commands are input, they appear on the display window from the left. Computational results, however, are displayed from the right.*

■ Corrections

- To make corrections in a formula that is being input, use the [◁] and [▷] keys to move to the position of the error and press the correct keys.

Ex. To change an input of 122 to 123:

Keys	Display
[1] [2] [2]	122_
[◁]	12̲2
[3]	123_

Ex. To change an input of cos60 to sin60 :

Keys	Display
[cos] [6] [0]	cos 60_
[◁] [◁] [◁]	c̲os 60
[sin]	sin 6̲0

** If, after making corrections, input of the formula is complete, the answer can be obtained by pressing [EXE]. If, however, more is to be added to the formula, advance the cursor using the [▷] key to the end of the formula for input.*

- If an unnecessary character has been included in a formula, use the [◁] and [▷] keys to move to the position of the error and press the [DEL] key. Each press of [DEL] will delete one command (one step).

Ex. To correct an input of 369××2 to 369×2:

Keys	Display
[3] [6] [9] [×] [×] [2]	369××2_
[◁] [◁] [DEL]	369×2

● If a character has been omitted from a formula, use the ◁ and ▷ keys to move to the position where the character should have been input, and press SHIFT followed by the INS key.

Ex. To correct an input of 2.36^2 to $\sin 2.36^2$:

2 · 3 6 x^2
◁ ◁ ◁ ◁ ◁
SHIFT INS
sin

2.36^2_
2.36^2
$\boxed{2}.36^2$
sin $\boxed{2}.36^2$

** Pressing SHIFT INS specifies the insert mode and causes "[ ]" to appear on the display. Pressing keys while in this mode inserts the corresponding characters at the position of "[ ]".*
Press ◁, ▷, SHIFT INS, or EXE to cancel the insert mode.

■ Memory

This unit contains 6 memories. Memory names are A through F (1~6). Numeric values with 12 digits for a mantissa and 2 digits for an exponent can be stored.

Ex. To store 123.45 in memory A:

123.45 → ALPHA A
EXE

123.45→A_
123.45

Values are assigned to a memory using the → key followed by the memory name.

Ex. To store the sum of memory A + 78.9 in memory B:

ALPHA A + 78.9 → ALPHA B
EXE

A+78.9→B_
202.35

Ex. To add 74.12 to memory B:

ALPHA B + 74.12 → ALPHA B
EXE

B+74.12→B_
276.47

● To check the contents of a memory, press the name of the memory to be checked followed by EXE.

ALPHA A EXE

123.45

● To clear the contents of a memory (make them 0), proceed as follows:

Ex. To clear the contents of memory A only:

0 → ALPHA A EXE

0.

Ex. To clear the contents of all the memories:

SHIFT Mcl
EXE

Mcl_
0.

■ Answer (Ans) function

This unit has an answer function that stores the result of the most recent computation. Once a numeric value or numeric formula is entered and EXE is pressed, the result (the answer in the case of the numeric formula) is stored by this function. To recall the stored value, press the Ans key.
When Ans is pressed, "Ans" will appear on the display, and can be used in this form in subsequent calculations.
** Hereinafter, Ans will be referred to as the Ans memory.*

Ex. 123+456=579
789−579=210

1 2 3 + 4 5 6 EXE
7 8 9 − Ans
EXE

579.
789−Ans_
210.

Numeric values with 12 digits for a mantissa and 2 digits for an exponent can be stored in the Ans memory. The Ans memory is not erased even if the power of the unit is switched OFF. Each time EXE is pressed, the value in the Ans memory is replaced with the new value produced by the computation executed.
When a value is stored to another memory using the EXE key, that value is not stored in the Ans memory.

Ex. Perform computation 78+56=134, then store the value 123 to memory A:

7 8 + 5 6 EXE
Ans EXE ... Confirm Ans memory contents.
1 2 3 → ALPHA A EXE
Ans EXE

134.
134.
123.
134.

The Ans memory can be used in the same manner as the other memories, thus making it possible to use it in computation formulas. In multiplication operations, the [×] immediately before [Ans] can be omitted.

Ex. $15\times3=45$

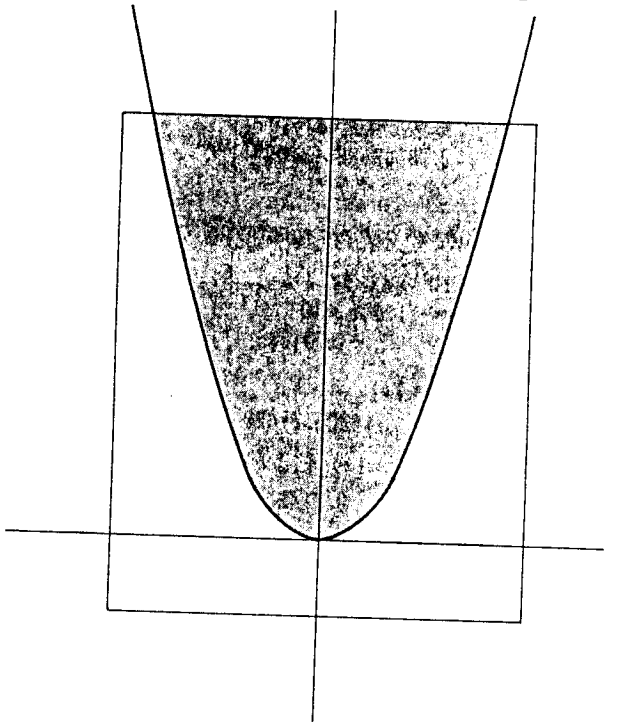
$78\times45-23=3487$

Keys	Display
[1] [5] [×] [3] [EXE]	45.
[7] [8] [Ans] [−] [2] [3] [EXE]	3487.

■ Auto power off function

The power of the unit is automatically switched off approximately 6 minutes after the last key operation (except during program computations). Once this occurs, power can be restored either by switching the power of the unit OFF and then ON again, or by pressing the [AC] key. (Numeric values in the memories, programs or computation modes are unaffected when power is switched off.)

2. MANUAL COMPUTATIONS



2-1 BASIC COMPUTATIONS

■ Arithmetic operations

- Arithmetic operations are performed by pressing the keys in the same order as noted in the formula.
- For negative values, press [(-)] before entering the value.

Example	Operation	Display
$23+4.5-53=-25.5$	23 [+] 4.5 [−] 53 [EXE]	-25.5
$56\times(-12)\div(-2.5)=268.8$	56 [×] [(-)] 12 [÷] [(-)] 2.5 [EXE]	268.8
$12369\times7532\times74103=6.903680613\times10^{12}$ (6903680613000)	12369 [×] 7532 [×] 74103 [EXE]	6.903680613E12
** Results greater than 10^{10} (10 billion) or less than 10^{-2} (0.01) are displayed in exponential form.*		
$(4.5\times10^{75})\times(-2.3\times10^{-79})=-1.035\times10^{-3}$ (−0.001035)	4.5 [EXP] 75 [×] [(-)] 2.3 [EXP] [(-)] 79 [EXE]	-1.035E-03
$(1\times10^{5})\div7=14285.71429$	1 [EXP] 5 [÷] 7 [EXE]	14285.71429
$(1\times10^{5})\div7-14285=0.7142857$	1 [EXP] 5 [÷] 7 [−] 14285 [EXE]	0.7142857
** Internal computations are computed in 12 digits for a mantissa, and the result is displayed rounded off to 10 digits.*		

- For mixed basic arithmetic operations, multiplication and division are given priority over addition and subtraction.

Example	Operation	Display
$3+\underline{5\times6}=33$	3 [+] 5 [×] 6 [EXE]	33.
$\underline{7\times8}-\underline{4\times5}=36$	7 [×] 8 [−] 4 [×] 5 [EXE]	36.
$1+2-\underline{3\times4\div5}+6=6.6$	1 [+] 2 [−] 3 [×] 4 [÷] 5 [+] 6 [EXE]	6.6

■ Parenthesis computations

Example	Operation	Display
$100-(2+3)\times4=80$	100 [−] [(] 2 [+] 3 [)] [×] 4 [EXE]	80.
$2+3\times(4+5)=29$	2 [+] 3 [×] [(] 4 [+] 5 [EXE]	29.
** Closed parentheses occuring immediately before operation of the [EXE] key may be omitted, no matter how many are required.*		
$(7-2)\times(8+5)=65$	[(] 7 [−] 2 [)] [(] 8+5 [EXE]	65.
** A multiplication sign (×) occuring immediately before an open parenthesis can be omitted.*		
$10-\{2+7\times(3+6)\}=-55$	10 [−] [(] 2 [+] 7 [(] 3 [+] 6 [EXE]	-55.
** Henceforth, abbreviated style will not be used in this manual.*		
$\frac{2\times3+4}{5}=(2\times3+4)\div5=2$	[(] 2 [×] 3 [+] 4 [)] [÷] 5 [EXE]	2.
$\frac{5\times6+6\times8}{15\times4+12\times3}=0.8125$	[(] 5 [×] 6 [+] 6 [×] 8 [)] [÷] [(] 15 [×] 4 [+] 12 [×] 3 [)] [EXE]	0.8125
$(1.2\times10^{19})-(2.5\times10^{20})\times\frac{3}{100})=4.5\times10^{18}$	1.2 [EXP] 19 [−] [(] 2.5 [EXP] 20 [×] 3 [÷] 100 [)] [EXE]	4.5E18
$\frac{6}{4\times5}=0.3$	6 [÷] [(] 4 [×] 5 [)] [EXE]	0.3
** The above is the same as* 6 [÷] 4 [÷] 5 [EXE] .		

■ Memory computations

- Numeric values stored in memory contain 12 digits for a mantissa.
- The contents of memories are not erased when power is switched OFF. They are cleared by pressing [SHIFT] followed by [Mcl] ([DEL] key) and then [EXE] .

Example	Operation	Display
9.874×7=69.118	9.874 [→] [ALPHA] [A] [EXE]	9.874
9.874×12=118.488	[ALPHA] [A] [×] 7 [EXE]	69.118
9.874×26=256.724	[ALPHA] [A] [×] 12 [EXE]	118.488
9.874×29=286.346	[ALPHA] [A] [×] 26 [EXE] [ALPHA] [A] [×] 29 [EXE]	256.724 286.346

**The [→] key is used to input numeric values in memory. (Clearing a memory before input is not required, because the previous value in the memory will be automatically replaced with the new value.)*

23+9=32	23 [+] 9 [→] [ALPHA] [B] [EXE]	32.
53−6=47	53 [−] 6 [EXE]	47.
−)45×2=90	[ALPHA] [B] [+] [Ans] [→] [ALPHA] [B] [EXE]	79.
99÷3=33	45 [×] 2 [EXE]	90.
Total 22	[ALPHA] [B] [−] [Ans] [→] [ALPHA] [B] [EXE]	−11.
	99 [÷] 3 [EXE]	33.
	[ALPHA] [B] [+] [Ans] [→] [ALPHA] [B] [EXE]	22.
12×(2.3+3.4)−5=63.4	2.3 [+] 3.4 [→] [ALPHA] [G] [EXE]	5.7
	12 [×] [ALPHA] [C] [−] 5 [EXE]	63.4
30×(2.3+3.4+4.5)−15 ×4.5=238.5	4.5 [→] [ALPHA] [H] [EXE]	4.5
	30 [×] [(] [ALPHA] [C] [+] [ALPHA] [D] [)] [−] 15 [ALPHA] [D] [EXE]	238.5

**Multiplication signs (×) immediately before memory names can be omitted.*

■ Specifying the number of decimal places, the number of significant digits and the exponent display

- To specify the number of decimal places, press [MODE] followed by [7] , a value indicating the number of places (0−9) and then [EXE]. (The indicator "Fix" will illuminate on the display.)
- To specify the number of significant digits, press [MODE] followed by [8] , a value indicating the number of significant digits (0−9 to set from 1 to 10 digits) and then [EXE] . (The indicator "Sci" will illuminate on the display.)
- Pressing the [ENG] key or [SHIFT] followed by [←] ([ENG] key) will cause the exponent display for the number being displayed to change in multiples of 3.
- The specified number of decimal places or number of significant digits will not be cancelled until another value or [MODE] [9] is specified using the sequence: [MODE], [9], [EXE]. (Specified values are not cancelled even if power is switched OFF or an other mode (besides [MODE] [9]) is specified.)
- Even if the number of decimal places and number of significant digits are specified, internal computations are performed in 12 digits for a mantissa, and the displayed value is stored in 10 digits. To convert these values to the specified number of decimal places and significant digits, press [SHIFT] followed by [Rnd] ([0] key) and then [EXE] .

** You cannot specify the display format (Fix, Sci) while the calculator is in the Base-n mode. Such specifications can only be made if you first exit the Base-n mode.*

Example	Operation	Display
100÷6=16.66666666···	100 [÷] 6 [EXE]	16.66666667
	[MODE] [7] [4] [EXE] (Four decimal places specified.)	16.6667
	[MODE] [9] [EXE] (Specification cancelled.)	16.66666667
	[MODE] [8] [5] [EXE] (Five significant digits specified.)	1.6667E 01
	[MODE] [9] [EXE] (Specification cancelled.)	16.66666667

** Values are displayed rounded off to the place specified.*

200÷7×14=400	MODE 7 3 EXE (Three decimal places specified.)	16.667
	200 ÷ 7 EXE	28.571
(Continues computation with 10-digit display.)	×	28.57142857×
	14 EXE	400.000
	If the same computation is performed with the specified number of digits:	
	200 ÷ 7 EXE	28.571
	(Value stored internally cut off at specified decimal place.) SHIFT Rnd EXE	28.571
	×	28.571×
	14 EXE	399.994
	MODE 9 EXE (Specification cancelled.)	399.994
123m×456=56088m	123 × 456 EXE	56088.
=56.088km	ENG	56.088E 03
78g×0.96=74.88g	78 × 0.96 EXE	74.88
=0.07488kg	SHIFT ENG	0.07488E 03

2-2 SPECIAL FUNCTIONS

■ Continuous computation function

Even if computations are concluded with the EXE key, the result obtained can be used for further computations. In this case, computations are performed with 10 digits for the mantissa which is displayed.

Ex. $3\times4=12$ **Continuing** $\div3.14=$

3 × 4 EXE
(Continuing) ÷ 3.14 EXE

12.
12.÷3.14_
3.821656051

Ex. To compute $1\div3\times3$

1 ÷ 3 × 3 EXE
1 ÷ 3 EXE
(Continuing) × 3 EXE

1.
0.3333333333
0.9999999999

This function can be used with memory and Type A functions (x^2, x^{-1}, $x!$: see page 38), and +, −, ${}_nP_r$, ${}_nC_r$, x^y, $\sqrt[x]{\ }$, °′″.

Ex. To store the result of 12×45 in memory C:

12 × 45 EXE
(Continuing) → ALPHA C EXE

540.
540.→C_
540.

Ex. To square the result of $78\div6$ (see page 38):

78 ÷ 6 EXE
(Continuing) x^2 EXE

13.
13.²_
169.

■ Replay function

- This function stores formulas that have been executed. After execution is complete pressing either the [⇨] or [⇦] key will display the formula executed.
  Pressing [⇨] will display the formula from the beginning, with the cursor located under the first character.
  Pressing [⇦] will display the formula from the end, with the cursor located at the space following the last character.
  Then using [⇨] and [⇦] to move the cursor, the formula can be checked and numeric values or commands can be changed for subsequent execution.

Ex.

123 [×] 456 [EXE]
[⇨]
[EXE]
[⇦]

56088.
123×456
56088.
123×456_

Ex. 4.12×3.58+6.4=21.1496
4.12×3.58−7.1=7.6496

4.12 [×] 3.58 [+] 6.4 [EXE]
[⇦]
[⇦] [⇦] [⇦] [⇦]
[−] 7.1
[EXE]

21.1496
4.12×3.58+6.4_
4.12×3.58±6.4
4.12×3.58−7.1_
7.6496

- If an error is generated during computation execution, an error check function eliminates the need to clear the error using [AC] and then restarting input from the beginning. Pressing either [⇨] or [⇦] will automatically move the cursor to the point in the formula that generated the error and display it.

Ex. When 14÷0×2.3 is mistakenly entered for 14÷10×2.3:

14 [÷] 0 [×] 2.3 [EXE]
[⇨] (or [⇦])

Ma ERROR
14÷0×2.3

Here an error occurs.

[⇦] [SHIFT] [INS] 1
[EXE]

14÷10×2.3
3.22

**As with the number of input characters (see page 18), the replay function can accept input up to 60 steps.*
**The replay function is cleared when the [AC] key is pressed, when power is switched OFF or when the mode is changed.*

■ Multistatement function

- The multistatement function (using colons to separate formulas or statements) available in program computations can also be used for manual computations.
- The multistatement function allows formulas to be separated by colons to make consecutive, multiple statement computations possible.
- When [EXE] is pressed to execute a formula input using the multistatement format, the formula is executed in order from the beginning.
- Inputting "◢" ([SHIFT] [:]) in place of the colon will display the computational result up to that point during execution.

Ex. 6.9×123=848.7
123÷3.2=38.4375

123 [→] [ALPHA] [A] [:]
6.9 [×] [ALPHA] [A] [SHIFT] [◢]
[ALPHA] [A] [÷] 3.2 [EXE]
[EXE]

Illuminated during display caused by "◢".

Disp 848.7
38.4375

**Even if "◢" is not input at the end of a formula, the final result will be displayed.*
**Consecutive computations using multistatements cannot be performed.*

123×456:+5
Invalid

2-3 FUNCTIONAL COMPUTATIONS

■ Angular measurement units

- The unit of angular measurement (degrees, radians, grads) is set by pressing [MODE] followed by a value from 4 through 6 and then [EXE].
- The numeric value from 4 through 6 specifies degrees, radians and grads respectively.
- Once a unit of angular measurement is set, it remains in effect until a new unit is set. Settings are not cleared when power is switched OFF.
- You cannot specify the unit of angular measurement (degrees, radians, grads) while the calculator is in the Base-n mode. Such specifications can only be made if you first exit the Base-n mode.

Example	Operation	Display
Conversion of 4.25 rad to degrees	[MODE] [4] [EXE] →"D" 4.25 [SHIFT] [MODE] [5] [EXE]	243.5070629
Conversion of 1.23 grad. to radians	[MODE] [5] [EXE] →"R" 1.23 [SHIFT] [MODE] [6] [EXE]	0.01932079482
Conversion of 7.89 degrees to grads	[MODE] [6] [EXE] →"G" 7.89 [SHIFT] [MODE] [4] [EXE]	8.766666667
Result displayed in degrees 47.3°+82.5rad= 4774.20181	[MODE] [4] [EXE] →"D" 47.3 [+] 82.5 [SHIFT] [MODE] [5] [EXE]	4774.20181
12.4°+8.3rad−1.8gra= 486.33497	12.4 [+] 8.3 [SHIFT] [MODE] [5] [−] 1.8 [SHIFT] [MODE] [6] [EXE]	486.33497
Result displayed in radians 24°6'31''+85.34rad= 85.76077464	[MODE] [5] [EXE] →"R" 24 [°'"] 6 [°'"] 31 [°'"] [SHIFT] [MODE] [4] [+] 85.34 [EXE]	85.76077464
Result displayed in grads 36.9°+41.2rad= 2663.873462	[MODE] [6] [EXE] →"G" 36.9 [SHIFT] [MODE] [4] [+] 41.2 [SHIFT] [MODE] [5] [EXE]	2663.873462

■ Trigonometric functions and inverse trigonometric functions

- Be sure to set the unit of angular measurement before performing trigonometric function and inverse trigonometric function computations.

Example	Operation	Display
sin63°52'41''= 0.897859012	[MODE] [4] [EXE] →"D" [sin] 63 [°'"] 52 [°'"] 41 [°'"] [EXE]	0.897859012
$\cos(\frac{\pi}{3}\text{rad})=0.5$	[MODE] [5] [EXE] →"R" [cos] [(] [SHIFT] [π] [÷] 3 [)] [EXE]	0.5
tan(−35gra)= −0.6128007881	[MODE] [6] [EXE] →"G" [tan] [(−)] 35 [EXE]	−0.6128007881
2·sin45°×cos65°= 0.5976724775	[MODE] [4] [EXE] →"D" 2 [×] [sin] 45 [×] [cos] 65 [EXE] Can be omitted.	0.5976724775
$\sin^{-1}0.5=30°$ (Determine the value of x when $\sin x=0.5$.)	[SHIFT] [sin⁻¹] 0.5 [EXE] Can be entered as .5	30.
$\cos^{-1}\frac{\sqrt{2}}{2}=$ 0.7853981634rad $=\frac{\pi}{4}$rad	[MODE] [5] [EXE] →"R" [SHIFT] [cos⁻¹] [(] [√] 2 [÷] 2 [)] [EXE] [÷] [SHIFT] [π] [EXE]	0.7853981634 0.25
tan⁻¹0.741= 36.53844577° =36°32'18.4"	[MODE] [4] [EXE] →"D" [SHIFT] [tan⁻¹] 0.741 [EXE] [SHIFT] [°'"]	36.53844577 36°32'18.4"

**If the total number of digits for degrees/minutes/seconds exceeds eleven digits, the high-order values (degrees and minutes) are given display priority, and any lower-order values are not displayed. However, the entire value is stored within the unit as a decimal value.*

2.5×(sin⁻¹0.8− cos⁻¹0.9)=68°13'13.53"	2.5 [×] [(] [SHIFT] [sin⁻¹] 0.8 [−] [SHIFT] [cos⁻¹] 0.9 [)] [EXE] [SHIFT] [°'"]	68°13'13.53"
sin18°×cos0.25rad= 0.2994104044	[sin] 18 [×] [cos] 0.25 [SHIFT] [MODE] [5] [EXE]	0.2994104044

**The above is computed in radians, and is the same as* [sin] 18 [SHIFT] [MODE] [4] [×] [cos] 0.25 [EXE].

■ Logarithmic and exponential functions

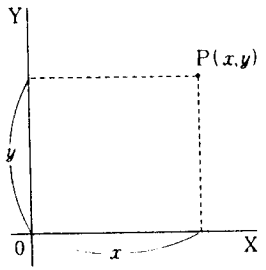
Example	Operation	Display
$\log 1.23 (\log_{10} 1.23) =$ $8.990511144 \times 10^{-2}$	log 1.23 EXE	0.08990511144
$\ln 90 (\log_e 90) =$ 4.49980967	ln 90 EXE	4.49980967
$\log 456 \div \ln 456 =$ 0.4342944819 (log/ln ratio = constant M)	log 456 ÷ ln 456 EXE	0.4342944819
$10^{1.23} = 16.98243652$ (To obtain the antilogarithm of common logarithm 1.23)	SHIFT 10^x 1.23 EXE	16.98243652
$e^{4.5} = 90.0171313$ (To obtain the antilogarithm of natural logarithm 4.5)	SHIFT e^x 4.5 EXE	90.0171313
$10^4 \cdot e^{-4} + 1.2 \cdot 10^{2.3}$ $= 422.5878667$	SHIFT 10^x 4 × SHIFT e^x (−) 4 + 1.2 × SHIFT 10^x 2.3 EXE	422.5878667
$5.6^{2.3} = 52.58143837$	5.6 x^y 2.3 EXE	52.58143837
$\sqrt[7]{123} (= 123^{\frac{1}{7}}) =$ 1.988647795	7 $\sqrt[x]{}$ 123 EXE	1.988647795
$(78-23)^{-12} =$ $1.305111829 \times 10^{-21}$	(78 − 23) x^y (−) 12 EXE	1.305111829E−21
$2 + 3 \times \sqrt[3]{64} - 4 = 10$	2 + 3 × 3 $\sqrt[x]{}$ 64 − 4 EXE	10.
* x^y and $\sqrt[x]{}$ given computation priority over × and ÷.		
$2 \times 3.4^{(5+6.7)} =$ 3306232.001	2 × 3.4 x^y (5 + 6.7) EXE	3306232.001

■ Hyperbolic functions and inverse hyperbolic functions

Example	Operation	Display
$\sinh 3.6 = 18.28545536$	hyp sin 3.6 EXE	18.28545536
$\cosh 1.23 = 1.856761057$	hyp cos 1.23 EXE	1.856761057
$\tanh 2.5 = 0.9866142981$	hyp tan 2.5 EXE	0.9866142981
$\cosh 1.5 - \sinh 1.5 =$ $0.2231301602 = e^{-1.5}$	hyp cos 1.5 − hyp sin 1.5 EXE (Continuing) ln Ans EXE	0.2231301602 −1.5
(Proof of $\cosh x \pm \sinh x = e^{\pm x}$)		
$\sinh^{-1} 30 = 4.094622224$	SHIFT hyp sin^-1 30 EXE	4.094622224
$\cosh^{-1} (\frac{20}{15}) =$ 0.7953654612	SHIFT hyp cos^-1 (20 ÷ 15) EXE	0.7953654612
Determine the value of x when $\tanh 4x = 0.88$		
$x = \frac{\tanh^{-1} 0.88}{4} =$ 0.3439419141	SHIFT hyp tan^-1 0.88 ÷ 4 EXE	0.3439419141
$\sinh^{-1} 2 \times \cosh^{-1} 1.5 =$ 1.389388923	SHIFT hyp sin^-1 2 × SHIFT hyp cos^-1 1.5 EXE	1.389388923
$\sinh^{-1} (\frac{2}{3}) + \tanh^{-1} (\frac{4}{5}) =$ 1.723757406	SHIFT hyp sin^-1 (2 ÷ 3) + SHIFT hyp tan^-1 (4 ÷ 5) EXE	1.723757406

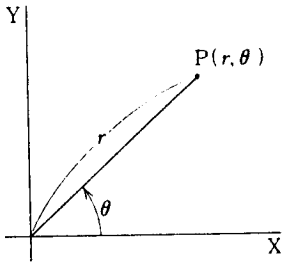
■ Coordinate transformation

● Rectangular coordinates



Pol →
← Rec

● Polar coordinates



● Computation results are stored in memories E and F. (Contents of memory E displayed.)

Pol → E = r, F = θ

Rec → E = x, F = y

● With polar coordinates, θ can be computed within a range of $-180° < \theta \leqq 180°$. (The computation range is the same with radians or grads.)

Example	Operation	Display
If $x=14$ and $y=20.7$, what are r and θ?	MODE 4 EXE →"D" SHIFT Pol(14 SHIFT , 20.7) EXE	24.98979792(r)
	(Continuing) ALPHA F EXE SHIFT °,,,	55°55'42.2"(θ)
If $x=7.5$ and $y=-10$, what are r and θ rad?	MODE 5 EXE →"R" SHIFT Pol(7.5 SHIFT , (−) 10) EXE	12.5(r)
	(Continuing) ALPHA F EXE	−0.927295218(θ)
If $r=25$ and $\theta=56$, what are x and y?	MODE 4 EXE →"D" SHIFT Rec 25 SHIFT , 56) EXE	13.97982259(x)
	(Continuing) ALPHA F EXE	20.72593931(y)
If $r=4.5$ and $\theta=\frac{2}{3}\pi$ rad, what are x and y?	MODE 5 EXE →"R" SHIFT Rec 4.5 SHIFT , (2 ÷ 3 × SHIFT π)) EXE	−2.25(x)
	(Continuing) ALPHA F EXE	3.897114317(y)

■ Permutation and combination

● Total number of permutations

$$nPr = \frac{n!}{(n-r)!}$$

● Total number of combinations

$$nCr = \frac{n!}{r!(n-r)!}$$

Example	Operation	Display
Taking any four out of ten items and arranging them in a row, how many different arrangements are possible? ${}_{10}P_4 = 5040$	10 SHIFT nPr 4 EXE	5040.
Using any four numbers from 1 to 7, how many four-digit even numbers can be formed if none of the four digits consist of the same number ? (3/7 of the total number of permutations will be even.) ${}_7P_4 \times \frac{3}{7} = 360$	7 SHIFT nPr 4 × 3 ÷ 7 EXE	360.
If any four items are removed from 10 items, how many different combinations are possible? ${}_{10}C_4 = 210$	10 SHIFT nCr 4 EXE	210.
If 5 class officers are being selected for a class of 15 boys and 10 girls, how many combinations are possible? At least one girl must be included in each group. ${}_{25}C_5 - {}_{15}C_5 = 50127$	25 SHIFT nCr 5 − 15 SHIFT nCr 5 EXE	50127.

■ Other functions ($\sqrt{\ }$, x^2, x^{-1}, $x!$, $\sqrt[3]{\ }$, RAN#, ABS, INT, FRAC)

Example	Operation	Display
$\sqrt{2}+\sqrt{5}=3.65028154$	[√] 2 [+] [√] 5 [EXE]	3.65028154
$2^2+3^2+4^2+5^2=54$	2 [x^2] [+] 3 [x^2] [+] 4 [x^2] [+] 5 [x^2] [EXE]	54.
$\frac{1}{\frac{1}{3}-\frac{1}{4}}=12$	[(] 3 [x^{-1}] − 4 [x^{-1}] [)] [x^{-1}] [EXE]	12.
$8!(=1\times2\times3\times\cdots\cdots\times8)$ $=40320$	8 [SHIFT] [$x!$] [EXE]	40320.
$\sqrt[3]{36\times42\times49}=42$	[SHIFT] [$\sqrt[3]{\ }$] [(] 36 [×] 42 [×] 49 [)] [EXE]	42.
Random number generation (pseudorandom number from 0.000 to 0.999)	[SHIFT] [Ran#] [EXE]	(Ex) 0.792
$\sqrt{13^2-5^2}+\sqrt{3^2+4^2}=17$	[√] [(] 13 [x^2] − 5 [x^2] [)] [+] [√] [(] 3 [x^2] [+] 4 [x^2] [)] [EXE]	17.
$\sqrt{1-\sin^2 40^\circ}=$ $0.7660444431=\cos 40^\circ$	[MODE] [4] [EXE] →"D" [√] [(] 1 − [(] [sin] 40 [)] [x^2] [)] [EXE]	0.7660444431
(Proof of $\cos\theta=\sqrt{1-\sin^2\theta}$)	(Continuing) [SHIFT] [cos⁻¹] [Ans] [EXE]	40.
$\frac{1}{2!}+\frac{1}{4!}+\frac{1}{6!}+\frac{1}{8!}=$ 0.5430803571	2 [SHIFT] [$x!$] [x^{-1}] [+] 4 [SHIFT] [$x!$] [x^{-1}] [+] 6 [SHIFT] [$x!$] [x^{-1}] [+] 8 [SHIFT] [$x!$] [x^{-1}] [EXE]	0.5430803571
What is the absolute value of the common logarithm of $\frac{3}{4}$? $\left\|\log\frac{3}{4}\right\|=0.1249387366$	[SHIFT] [Abs] [log] [(] 3 ÷ 4 [)] [EXE]	0.1249387366

Example	Operation	Display
What is the integer part of $\frac{7800}{96}$?	[SHIFT] [Int] [(] 7800 ÷ 96 [)] [EXE]	81.
What is the fraction part of $\frac{7800}{96}$?	[SHIFT] [Frac] [(] 7800 ÷ 96 [)] [EXE]	0.25
What is the aliquot part of 2512549139÷2141?	2512549139 ÷ 2141 [EXE] [SHIFT] [Frac] [(] 2512549139 ÷ 2141 [)] [EXE]	1173540. 0.99953

2-4 ENGINEERING SYMBOL COMPUTATIONS

- Engineering computations using ENG can be performed.
- The ENG mode (ENG lights on display) is specified by pressing [MODE] [·]. Pressing [MODE] [·] again cancels the ENG mode and causes ENG to disappear from the display.

Operation	Unit	Unit symbol
[ALPHA] [k] (=[x^{-1}])	10^{3}	k (kilo)
[ALPHA] [M] (=[°,,,])	10^{6}	M (mega)
[ALPHA] [G] (=[hyp])	10^{9}	G (giga)
[ALPHA] [T] (=[sin])	10^{12}	T (tera)
[ALPHA] [P] (=[cos])	10^{15}	P (peta)
[ALPHA] [E] (=[tan])	10^{18}	E (exa)
[ALPHA] [m] (=[:])	10^{-3}	m (milli)
[ALPHA] [μ] (=[ENG])	10^{-6}	μ (micro)
[ALPHA] [n] (=[√‾])	10^{-9}	n (nano)
[ALPHA] [p] (=[x^{2}])	10^{-12}	p (pico)
[ALPHA] [f] (=[log])	10^{-15}	f (femto)
[ALPHA] [a] (=[ln])	10^{-18}	a (atto)

Example	Operation	Display
999k (kilo)+25k (kilo)= 1.024M (mega)	999 [ALPHA] [k] [+] 25 [ALPHA] [k] [EXE]	1.024M
100m (milli)×5 μ (micro)= 500n (nano)	100 [ALPHA] [m] [×] 5 [ALPHA] [μ] [EXE]	500.n
9÷10=0.9=900m (milli)	9 [÷] 10 [EXE]	900.m
	[SHIFT] [ENG]	0.9
	[ENG]	900.m
1E (exa)×1000=10^{21}	1 [ALPHA] [E] [×] 1000 [EXE]	1.E 21

** The display range when using ENG symbols is from 1 through 1000 for the mantissa, and from −18 through 18 for the exponent.*

2-5 BINARY, OCTAL, DECIMAL, HEXADECIMAL COMPUTATIONS

- Binary, octal, decimal and hexadecimal computations, conversions and logical operations are performed in the Base-n mode (press [MODE] [−])
- The number system (2, 8, 10, 16) is set by respectively pressing [Bin] , [Oct] , [Dec] or [Hex] , followed by [EXE] .
- Number systems are specified for specific values by pressing [SHIFT] , then the number system designator ([b] , [o] , [d] or [h]), immediately followed by the value.
- General function computations cannot be performed in the Base-n mode.
- Only integers can be handled in the Base-n mode. If a computation produces a result that includes a decimal value, the decimal portion is cut off.
- Computations can be handled up to 32 bits.

Binary	Up to 32 digits (Each press of [Block] switches display between high-order 16 digits and low-order 16 digits.)
Octal	Up to 11 digits
Decimal	Up to 10 digits
Hexadecimal	Up to 8 digits

- The total range of numbers handled in this mode is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. If values not valid for the particular number system are used, attach the corresponding designator (b, o, d or h), or an error message will appear.

Valid values

Binary	0, 1
Octal	0, 1, 2, 3, 4, 5, 6, 7
Decimal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

- Negative numbers in binary, octal and hexadecimal are expressed as twos complements.
- To distinguish the A, B, C, D, E, F used in the hexadecimal system from standard letters they appear as: 𝔸, 𝔹, ℂ, 𝔻, 𝔼, 𝔽.

● Computation range (in Base-n mode)

Binary	Positive: $1111111111111111111111111111111 \geqq x \geqq 0$ Negative: $1111111111111111111111111111111 \geqq x \geqq 1000000000000000000000000000000$
Octal	Positive: $17777777777 \geqq x \geqq 0$ Negative: $37777777777 \geqq x \geqq 20000000000$
Decimal	Positive: $2147483647 \geqq x \geqq 0$ Negative: $-1 \geqq x \geqq -2147483648$
Hexadecimal	Positive: $7FFFFFFF \geqq x \geqq 0$ Negative: $FFFFFFFF \geqq x \geqq 80000000$

● You cannot specify the unit of angular measurement (degrees, radians, grads) or the display format (Fix, Sci) while the calculator is in the Base-n mode. Such specifications can only be made if you first exit the Base-n mode.

■ Binary, octal, decimal, hexadecimal conversions

Example	Operation	Display
	[MODE] [—] →"Base-n"	
What are the decimal values for $2A_{16}$ and 274_8?	[Dec] [EXE] →"Dec" [SHIFT] [h] 2A [EXE] [SHIFT] [o] 274 [EXE]	42. 188.
What are the hexadecimal values for 123_{10} and 1010_2?	[Hex] [EXE] →"Hex" [SHIFT] [d] 123 [EXE] [SHIFT] [b] 1010 [EXE]	0000007B 0000000A
What are the octal values for 15_{16} and 1100_2?	[Oct] [EXE] →"Oct" [SHIFT] [h] 15 [EXE] [SHIFT] [b] 1100 [EXE]	00000000025 00000000014
What are the binary values for 36_{10} and $2C3B7_{16}$	[Bin] [EXE] →"Bin" [SHIFT] [d] 36 [EXE] [SHIFT] [h] 2C3B7 [EXE] (Continuing) [Block]	0000000000100100 1100001110110111 0000000000000010 (=10110000111011 0111)

■ Negative expressions

Example	Operation	Display
	[MODE] [—] →"Base-n"	
How is 110010_2 expressed as a negative?	[Bin] [EXE] →"Bin" [Neg] 110010 [EXE] [Block]	1111111111001110 1111111111111111 (=11111111111111111111111111001110)
How is 72_8 expressed as a negative?	[Oct] [EXE] →"Oct" [Neg] 72 [EXE]	37777777706
How is $3A_{16}$ expressed as a negative?	[Hex] [EXE] →"Hex" [Neg] 3A [EXE]	FFFFFFC6

■ Basic arithmetic operations using binary, octal, decimal and hexadecimal values

Example	Operation	Display
	[MODE] [—] →"Base-n"	
$10111_2+11010_2=110001_2$	[Bin] [EXE] →"Bin" 10111 [+] 11010 [EXE]	0000000000110001
$B47_{16}-DF_{16}=A68_{16}$	[Hex] [EXE] →"Hex" B47 [—] DF [EXE]	00000A68
$123_8\times ABC_{16}=37AF4_{16}$ $=228084_{10}$	[SHIFT] [o] 123 [×] ABC [EXE] [Dec] [EXE]	00037AF4 228084
$1F2D_{16}-100_{10}=7881_{10}$ $=1EC9_{16}$	[SHIFT] [h] 1F2D [—] 100 [EXE] [Hex] [EXE]	7881 00001EC9
$7654_8\div 12_{10}$ $=334.3333333_{10}$ $=516_8$	[Dec] [EXE] →"Dec" [SHIFT] [o] 7654 [÷] 12 [EXE] [Oct] [EXE]	334 00000000516

**Computation results are displayed with the decimal portion cut off.*

$1234_{10}+1EF_{16}\div24_8=2352_8$	[SHIFT] [d] 1234 [+] [SHIFT] [h] 1EF [÷] 24 [EXE]	00000002352
$=1258_{10}$	[Dec] [EXE]	1258

**For mixed basic arithmetic operations, multiplication and division are given computation priority over addition and subtraction.*

■ Logical operations

Logical operations are performed through logical product (AND), logical sum (OR) and, exclusive logical sum (XOR) and negation (NOT).

Example	Operation	Display
	[MODE] [−] →"Base-n"	
19_{16} AND $1A_{16}=18_{16}$	[Hex] [EXE] →"Hex" 19 [and] 1A [EXE]	00000018
1110_2 AND $36_8=1110_2$	[Bin] [EXE] →"Bin" 1110 [and] [SHIFT] [o] 36 [EXE]	0000000000001110
23_8 OR $61_8=63_8$	[Oct] [EXE] →"Oct" 23 [or] 61 [EXE]	00000000063
120_{16} OR $1101_2=12D_{16}$	[Hex] [EXE] →"Hex" 120 [or] [SHIFT] [b] 1101 [EXE]	0000012D
1010_2 AND (A_{16} OR 7_{16}) $=1010_2$	[Bin] [EXE] →"Bin" 1010 [and] [(] [SHIFT] [h] A [or] [SHIFT] [h] 7 [)] [EXE]	0000000000001010
5_{16} XOR $3_{16}=6_{16}$	[Hex] [EXE] → "Hex" 5 [SHIFT] [Xor] 3 [EXE]	00000006
Negation of 1234_8	[Oct] [EXE] →"Oct" [NOT] 1234 [EXE]	37777776543
Negation of $2FFFED_{16}$	[Hex] [EXE] →"Hex" [NOT] 2FFFED [EXE]	FFD00012

2-6 COMPLEX NUMBER COMPUTATIONS

- The complex number computations (CMPLX) mode is specified by pressing [MODE] [O].
- Pressing [i] ([ENG]) displays the imaginary number for an entered value.
- The real number part and imaginary number part are separated by "+" or "−" on the display, with the symbol " i " following the imaginary part.
- "1 i" displays " i " only.
- Computations are performed using the two memories A and B.
- In this mode, the numeric stack has three levels, while the command stack has 14 levels.
- Fix, Sci and Norm may also be specified while in the CMPLX mode for both real number part and imaginary number part.
- The replay function can be used in the CMPLX mode by pressing a cursor key ([⇦] or [⇨]) after [SHIFT]. ([SHIFT] is not necessary when an error is generated in the CMPLX mode.)
- ENG symbols may also be used in the CMPLX mode.

■ General computations

Example	Operation	Display
$4+8i+(6-2i)=10+6i$	4 [+] 8 [i] [+] 6 [−] 2 [i] [EXE]	10+6 i
$12i-34i=-22i$	12 [i] [−] 34 [i] [EXE]	−22 i
$8\times2i-18\div3i=22i$	8 [×] 2 [i] [−] 18 [÷] 3 [i] [EXE]	22 i
**Multiplication and division are given priority over addition and subtraction in mixed computations.*		
$2\times(3+4i)=6+8i$	2 [×] [(] 3 [+] 4 [i] [)] [EXE]	6+8 i
$8+(\underline{4+8i})=12+8i$	4 [+] 8 [i] [→] [ALPHA] [A] [EXE]	
	8 [+] [ALPHA] [A] [EXE]	12+8 i
$15-(\underline{4+8i})=11-8i$	15 [−] [ALPHA] [A] [EXE]	11−8 i
$3\times(\underline{4+8i})=12+24i$	3 [×] [ALPHA] [A] [EXE]	12+24 i
$(1.234+5.67i)\div80=$	[(] 1.234 [+] 5.67 [i] [)] [÷]	
$0.015425+0.070875i$	80 [EXE]	0.015425+0.07087
	[⇨] [⇨]	015425+0.070875 i
**Use [⇨] and [⇦] to view portions of a value which exceed the physical size of the display.*		

$(4+i)\div 3=$ $1.33333\cdots+0.33333\cdots i$	(4 + i) ÷ 3 EXE MODE 7 3 EXE MODE 8 2 EXE MODE 9 EXE	1.333333333+0.3$\dot{3}$ 1.333+0.333 i 1.3E00+3.3E−01 i 1.333333333+0.3$\dot{3}$

■ Other computations

(Absolute values, arguments, conjugate complex numbers, real number extraction, imaginary number extraction, squares, reciprocals)

- With complex numbers, absolute values (Abs), arguments. (Arg), conjugate complex numbers (conjg), real number extraction (Re·P), and imaginary number extraction (Im·P) can all be obtained.

Example	Operation	Display
Absolute value for $7+4i$	SHIFT Abs (7 + 4 i) EXE	8.062257748
Argument for $5+8i$	MODE 4 EXE SHIFT Arg (5 + 8 i) EXE	57.99461679
Argument for $3.2-4.8i$ (in radians)	MODE 5 EXE SHIFT Arg (3.2 − 4.8 i) EXE	−0.9827937233
Conjugate complex number for $8+5i$	SHIFT conjg (8 + 5 i) EXE	8−5 i
Conjugate complex number for $3\times(4-6i)$	SHIFT conjg (3 × (4 − 6 i)) EXE	12+18 i
Real number part of $(5+2i)\div 4$	SHIFT Re·P ((5 + 2 i) ÷ 4) EXE	1.25
Imaginary number part of $(5-2i)\div 4$	SHIFT Im·P ((5 + 2 i) ÷ 4) EXE	−0.5
$(2+5i)^2=-21+20i$	(2 + 5 i) x^2 EXE	−21+20 i
$1\div(4-2i)=0.2+0.1i$	(4 − 2 i) x^{-1} EXE	0.2+0.1 i

* Function commands beside ENG, $\overleftarrow{\text{ENG}}$ and $\sqrt[3]{}$ may be executed in the CMPLX mode, but such operation are only valid for the real number part or the coefficient in relation to the i of the imaginary number part.

Example:

$\sqrt{}$ 2 + (sin 30) i EXE → OK

log (1 + 2 i) EXE → Ma ERROR

2-7 STATISTICAL COMPUTATIONS

■ Standard deviation

- Standard deviation computations are performed in the SD mode. (Press MODE × , and "SD" illuminates on display.)
- Before beginning computations, the tabulation memories are cleared by pressing SHIFT followed by Scl (AC key) and then EXE .
- Individual data is input using DT ($\sqrt{}$ key).
- Multiple data of the same value can be input either by repeatedly pressing DT or by entering the data, pressing SHIFT , followed by ; , a value that represents the number of times the data is repeated, and then DT .

* *DT cannot be entered consectively in a program.*

- Standard deviation

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n}} = \sqrt{\frac{\Sigma x^2-(\Sigma x)^2/n}{n}}$$

Using the entire data of a finite population to determine the standard deviation for the population.

$$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} = \sqrt{\frac{\Sigma x^2-(\Sigma x)^2/n}{n-1}}$$

Using sample data for a population to determine the standard deviation for the population.

- Mean

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{\Sigma x}{n}$$

* *The values for n, Σx, and Σx² are stored in memories C, B, and A respectively, and can be obtained by pressing ALPHA followed by the memory name and then EXE (i,e, ALPHA C EXE).*

Example	Operation	Display
Data 55,54,51,55, 53,53,54,52	MODE × →"SD" SHIFT Scl EXE (Memory clear) 55 DT 54 DT 51 DT 55 DT 53 DT DT 54 DT 52 DT	52.
** Results can be obtained in any order desired.*		
($x\sigma_n$ = 2 $x\sigma_{n-1}$ = B $\bar{x}$ = 1)	(Standard deviation σ_n) SHIFT $x\sigma_n$ EXE	1.316956719
	(Standard deviation σ_{n-1}) SHIFT $x\sigma_{n-1}$ EXE	1.407885953
	(Mean $\bar{x}$) SHIFT $\bar{x}$ EXE	53.375
	(Number of data n) ALPHA C EXE	8.
	(Sum total Σx) ALPHA B EXE	427.
	(Sum of squares Σx^2) ALPHA A EXE	22805.
What is deviation of the unbiased variance, the difference between each datum, and the mean of the above data?	(Continuing) SHIFT $x\sigma_{n-1}$ x^2 EXE	1.982142857
	55 − SHIFT $\bar{x}$ EXE	1.625
	54 − SHIFT $\bar{x}$ EXE	0.625
	51 − SHIFT $\bar{x}$ EXE	−2.375
	⋮	⋮
What is $\bar{x}$ and $x\sigma_{n-1}$ for the following table?	SHIFT Scl EXE	
	110 SHIFT ; 10 DT	110.
	130 SHIFT ; 31 DT	130.
	150 SHIFT ; 24 DT	150.
	170 DT DT	170.
	190 DT DT DT	190.
	ALPHA C EXE	70.
	SHIFT $\bar{x}$ EXE	137.7142857
	SHIFT $x\sigma_{n-1}$ EXE	18.42898069

Class No.	Value	Frequency
1	110	10
2	130	31
3	150	24
4	170	2
5	190	3

*Erroneous data clearing/correction I (correct data operation: 51 DT)
① If 50 DT is entered, enter correct data after pressing CL (x^2 key).
② If 49 DT was input a number of entries previously, enter correct data after pressing 49 CL .

*Erroneous data clearing/correction II (correct data operation : 130 SHIFT ; 31 DT)
① If 120 SHIFT ; is entered, enter correct data after pressing AC .
② If 120 SHIFT ; 31 is entered, enter correct data after pressing AC .
③ If 120 SHIFT ; 30 DT is entered, enter correct data after pressing CI .
④ If 120 SHIFT ; 30 DT was entered previously, enter correct data after pressing 120 SHIFT ; 30 CL .

■ Regression computation

- Regression computations are performed in the LR mode. (Press MODE ÷ , and "LR" illuminates on display.)
- Before beginning computations, the tabulation memories are cleared by pressing SHIFT followed by Scl and then EXE .
- Individual data are entered as x data SHIFT , y data DT .
- Multiple data of the same value can be entered by repeatedly pressing DT . This operation can also be performed by entering x data SHIFT , y data SHIFT ; followed by a value representing the number of times the data is repeated, and then DT .

* DT *cannot be entered consecutively in a program.*

- The regression formula is $y=A+Bx$, and constant term A and regression coefficient B are computed using the following formulas:

Regression coefficient of regression formula

$$B = \frac{n\cdot\Sigma xy-\Sigma x\cdot\Sigma y}{n\cdot\Sigma x^2-(\Sigma x)^2}$$

Constant term of regression formula

$$A = \frac{\Sigma y-B\cdot\Sigma x}{n}$$

- Estimated values $\hat{x}$ and $\hat{y}$ based on the regression formula can be computed.
- The correlation coefficient r for input data can be computed using the following formula:

$$r = \frac{n\cdot\Sigma xy-\Sigma x\cdot\Sigma y}{\sqrt{\{n\cdot\Sigma x^2-(\Sigma x)^2\}\{n\cdot\Sigma y^2-(\Sigma y)^2\}}}$$

**The values for n, Σx, Σx^2, Σxy, Σy and Σy^2 are stored in memories C, B, A, F, E and D respectively, and can be obtained by pressing ALPHA followed by the memory name and then EXE (i.e. ALPHA C EXE).*

◆ Linear regression

Example	Operation	Display
● Temperature and the length of a steel bar Temp. / Length 10℃ / 1003mm 15 / 1005 20 / 1010 25 / 1011 30 / 1014 Using this table the regression formula and correlation coefficient can be obtained. Based on the coefficient formula, the length of the steel bar at 18℃ and the temperature at 1000mm can be estimated. Furthermore, the critical coefficient (r^2) and covariance $\left(\frac{\Sigma xy - n\cdot\bar{x}\cdot\bar{y}}{n-1}\right)$ can also be computed.	MODE ÷ →"LR" SHIFT Scl EXE (Memory clear) 10 SHIFT , 1003 DT 15 SHIFT , 1005 DT 20 SHIFT , 1010 DT 25 SHIFT , 1011 DT 30 SHIFT , 1014 DT (Constant term A) SHIFT A EXE (Regression coefficient B) SHIFT B EXE (Correlation coefficient r) SHIFT r EXE (Length at 18℃) 18 SHIFT $\hat{y}$ EXE (Temperature at 1000mm) 1000 SHIFT $\hat{x}$ EXE (Critical coefficient) SHIFT r x^2 EXE (Covariance) (ALPHA F — ALPHA C × SHIFT $\bar{x}$ × SHIFT $\bar{y}$) ÷ (ALPHA C — 1) EXE	10. 15. 20. 25. 30. 997.4 0.56 0.9826073689 1007.48 4.642857143 0.9655172414 35.

*Erroneous data clearing/correction (correct data operation : 10 SHIFT , 1003 DT)

① If 11 SHIFT , 1003 is entered, enter correct data after pressing AC.

② If 11 SHIFT , 1003 DT is entered, enter correct data after pressing CL

③ If 11 SHIFT , 1003 DT was entered previously, enter correct data after pressing 11 SHIFT , 1003 CL .

◆ Logarithmic regression

- The regression formula is $y=A+B\cdot\ln x$. Enter the x data as the logarithm (ln) of x, and the y data inputs the same as that for linear regression.
- The same operation as with linear regression can be used to obtain the regression coefficient and for making corrections. To obtain the estimated value $\hat{y}$, ln x SHIFT $\hat{y}$ EXE is used, and to obtain estimated value $\hat{x}$, y SHIFT $\hat{x}$ EXE SHIFT e^x Ans EXE is used.

Furthermore, Σx, Σx^2 and Σxy are obtained as $\Sigma\ln x$, $\Sigma(\ln x)^2$, and $\Sigma\ln xy$ respectively.

Example	Operation	Display
x_i / y_i 29 / 1.6 50 / 23.5 74 / 38.0 103 / 46.4 118 / 48.9 Through logarithmic regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, respective estimated values $\hat{y}$ and $\hat{x}$ can be obtained for $xi=80$ and $yi=73$ using the regression formula.	MODE ÷ →"LR" SHIFT Scl EXE ln 29 SHIFT , 1.6 DT ln 50 SHIFT , 23.5 DT ln 74 SHIFT , 38.0 DT ln 103 SHIFT , 46.4 DT ln 118 SHIFT , 48.9 DT (Constant term A) SHIFT A EXE (Regression coefficient B) SHIFT B EXE (Correlation coefficient r) SHIFT r EXE ($\hat{y}$ when $xi=80$) ln 80 SHIFT $\hat{y}$ EXE ($\hat{x}$ when $yi=73$) 73 SHIFT $\hat{x}$ EXE SHIFT e^x Ans EXE	3.36729583 3.912023005 4.304065093 4.634728988 4.770684624 −111.1283976 34.02014749 0.9940139464 37.94879482 224.1541314

◆ Exponential regression

- The regression formula is $y=A\cdot e^{B\cdot x}$ ($\ln y=\ln A+Bx$). Enter the y data as the logarithm of y(ln), and the x data the same as that for linear regression.
- Correction is performed the same as in linear regression. Constant term A is obtained by [SHIFT] [e^x] [SHIFT] [A] [EXE], estimated value $\hat{y}$ is obtained by x [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE], and estimated value $\hat{x}$ is obtained by [ln] y [SHIFT] [$\hat{x}$] [EXE]. Σy, Σy^2 and Σxy are obtained by $\Sigma\ln y$, $\Sigma(\ln y)^2$ and $\Sigma x\cdot\ln y$ respectively.

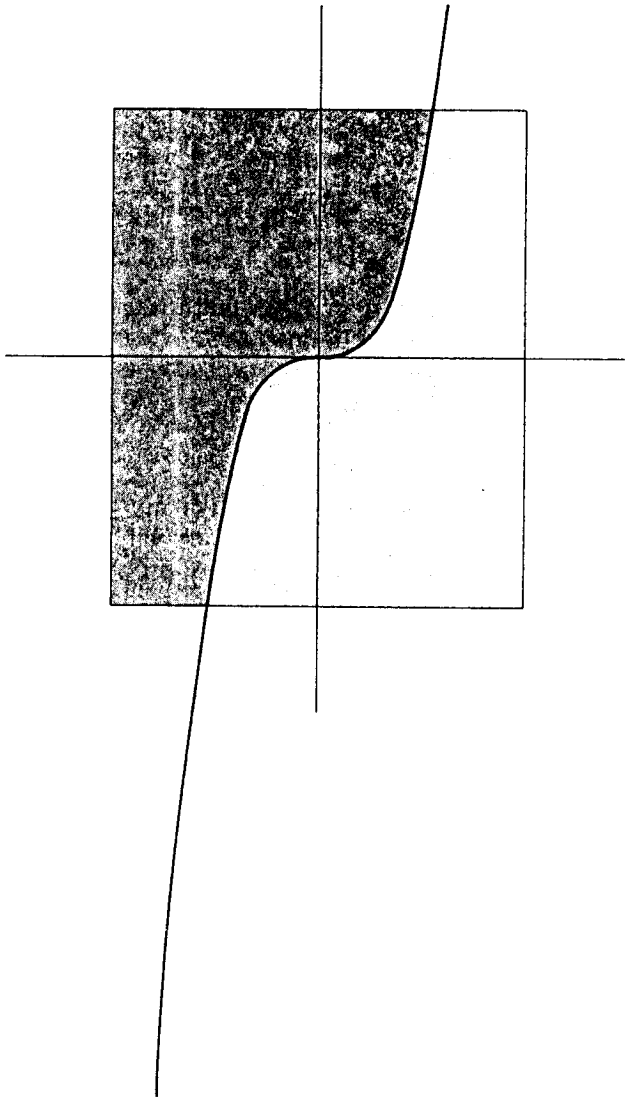
Example	Operation	Display
x_i | y_i 6.9 | 21.4 12.9 | 15.7 19.8 | 12.1 26.7 | 8.5 35.1 | 5.2	[SHIFT] [Scl] [EXE]	
	6.9 [SHIFT] [,] [ln] 21.4 [DT]	6.9
	12.9 [SHIFT] [,] [ln] 15.7 [DT]	12.9
	19.8 [SHIFT] [,] [ln] 12.1 [DT]	19.8
	26.7 [SHIFT] [,] [ln] 8.5 [DT]	26.7
	35.1 [SHIFT] [,] [ln] 5.2 [DT]	35.1
Through exponential regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, the regression formula is used to obtain the respective estimated value $\hat{y}$ and $\hat{x}$ when $xi=16$ and $yi=20$.	(Constant term A) [SHIFT] [e^x] [SHIFT] [A] [EXE]	30.49758742
	(Regression coefficient B) [SHIFT] [B] [EXE]	−0.04920370831
	(Correlation coefficient r) [SHIFT] [r] [EXE]	−0.9972473519
	($\hat{y}$ when $xi=16$) 16 [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	13.87915739
	($\hat{x}$ when $yi=20$) [ln] 20 [SHIFT] [$\hat{x}$] [EXE]	8.574868046

◆ Power regression

- The regression formula is $y=A\cdot x^B$ ($\ln y=\ln A+B\ln x$). Enter both data x and y as logarithms (ln).
- Correction is performed the same as in linear regression. Constant term A is obtained by [SHIFT] [e^x] [SHIFT] [A] [EXE], estimated value $\hat{y}$ is obtained by [ln] x [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE], and estimated value $\hat{x}$ is obtained by [ln] y [SHIFT] [$\hat{x}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]. Σx, Σx^2, Σy, Σy^2 and Σxy are obtained by $\Sigma\ln x$, $\Sigma(\ln x)^2$, $\Sigma\ln y$, $\Sigma(\ln y)^2$ and $\Sigma\ln x\cdot\ln y$ respectively.

Example	Operation	Display
xi | yi 28 | 2410 30 | 3033 33 | 3895 35 | 4491 38 | 5717	[SHIFT] [Scl] [EXE]	
	[ln] 28 [SHIFT] [,] [ln] 2410 [DT]	3.33220451
	[ln] 30 [SHIFT] [,] [ln] 3033 [DT]	3.401197382
	[ln] 33 [SHIFT] [,] [ln] 3895 [DT]	3.496507561
	[ln] 35 [SHIFT] [,] [ln] 4491 [DT]	3.555348061
	[ln] 38 [SHIFT] [,] [ln] 5717 [DT]	3.63758616
Through power regression of the above data, the regression formula and correlation coefficient are obtained. Furthermore, the regression formula is used to obtain the respective estimated values $\hat{y}$ and $\hat{x}$ when $xi=40$ and $yi=1000$.	(Constant term A) [SHIFT] [e^x] [SHIFT] [A] [EXE]	0.2388010829
	(Regression coefficient B) [SHIFT] [B] [EXE]	2.771866148
	(Correlation coefficient r) [SHIFT] [r] [EXE]	0.9989062562
	($\hat{y}$ when $xi=40$) [ln] 40 [SHIFT] [$\hat{y}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	6587.674743
	($\hat{x}$ when $yi=1000$) [ln] 1000 [SHIFT] [$\hat{x}$] [EXE] [SHIFT] [e^x] [Ans] [EXE]	20.26225659

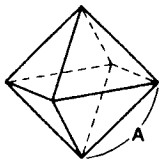
3.
PROGRAM COMPUTATIONS



3-1 WHAT IS A PROGRAM?

This unit has a built-in program feature that facilitates repeat computations. The program feature is used for the consecutive execution of formulas in the same way as the "multistatement" feature is used in manual computations. Programs will be discussed here with the aid of illustrative examples.

EXAMPLE:
Find the surface area and volume of a regular octahedron when the length of one side is given.



Length of one side (A)	Surface area (S)	Volume (V)
10cm	( )cm^2	( )cm^3
7	( )	( )
15	( )	( )

*Fill in the parentheses.

① Formulas

For a surface area S, volume V and one side A, S and V for a regular octahedron is defined as:

$$S=2\sqrt{3}A^2 \qquad V=\frac{\sqrt{2}}{3}A^3$$

② Programming

Creating a program based on computation formulas is known as "programming". Here a program will be created based upon the formulas given above. The basis of a program is manual computation, so first of all, consider the operational method used for manual computation.

Surface area (S): 2 [×] [√] 3 [×] Numeric value A [x²] [EXE]
Volume(V): [√] 2 [÷] 3 [×] Numeric value A [x^y] 3 [EXE]

In the above example, numeric value A is used twice, so it should make sense to store it in memory A before the computations.

Numeric value A → [ALPHA] [A] [EXE]
2 [×] [√] 3 [×] [ALPHA] [A] [x²] [EXE] ..... S
[√] 2 [÷] 3 [×] [ALPHA] [A] [x^y] 3 [EXE] V

With this unit, the operations performed for manual computations can be used as they are in a program. Once program execution starts, it will continue in order without stopping. Therefore, commands are required to request the input of data and to display results. The command to request data input is "?", while that to display results is "◢". A "?"within a program will cause execution to stop temporarily and a "?" to appear on the display as the unit waits for data input. This command cannot be used independently, and is used together with [→] as "[SHIFT] [?] [→] memory name". To store a numeric value in memory A, for example:

?→A

"A?" is displayed during output.

The"◢" command causes program execution to stop temporarily and the latest formula result or alphanumeric characters and symbols to be displayed. This command is used to mark positions in formulas where results are to be displayed. Since programs are ended and their final results displayed automatically, this command can be omitted at the end of a program.

Here these two commands will be used in the previously presented procedure:

[SHIFT] [?] [→] [ALPHA] [A] [:] 2 [×] [√] 3 [×] [ALPHA] [A] [x²] [SHIFT] [◢]
(Input to memory A; Display S)
[√] 2 [÷] 3 [×] [ALPHA] [A] [x^y] 3
(◢ omitted)

Now the program is complete.

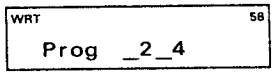
③ Program storage

The storage of programs is performed in the WRT mode which is specified by pressing [MODE] [2] ("WRT" illuminates on the display).

Operation	Display
[MODE] [2]	WRT D 100 Prog 1234

When [MODE] [2] are pressed, "WRT" appears at the upper left of the display to indicate that the unit is in the WRT mode. Then, the number of remaining steps (see page 66) is indicated at the upper right of the display. The number of remaining steps is decreased when programs are input. If no programs have been input, the number of usable steps should equal 100.

The larger figures located below indicate the program areas (see page 68). If the letters "Prog" are followed by the numbers 1 through 4 , it indicates that there are no programs stored in areas P1 through P4. The blinking 1 here indicates the current program area is P1.
Areas into which programs have already been stored are indicated by "—" instead of numbers.



Here the previously mentioned program will be stored to program area P0 (indicated by the blinking zero):

Operation	Display	
[EXE] (Writing starts)	WRT 0 _	The number at the upper right indicates the number of steps used in the currently specified program area.
[SHIFT] [?] [→] [ALPHA] [A] [:]	WRT 4 ?→A:_	
2 [×] [√] 3 [×] [ALPHA] [A] [x^2] [SHIFT] [◢]	WRT 12 ?→A:2X√3XA²◢_	When the number of input characters exceeds the capacity of the display, each subsequent input character will cause the displayed characters to shift to the left. This condition will be indicated by a "←".
[√] 2 [÷] 3 [×] [ALPHA] [A] [x^y] 3	WRT 20 √3XA²◢√2÷3XAx^y3_	

After these operations are complete, the program is stored.

*After the program is stored, press [MODE] [1] to return to the RUN mode.
*Note that the following functions cannot be used in a program:
[°'"], [$\overleftarrow{°'"}$], [ENG], [$\overleftarrow{ENG}$]

④ Program execution

Programs are executed in the RUN mode ([MODE] [1]). The program area to be executed is specified using the [Prog] key.

To execute P1: [Prog] 1 [EXE]
To execute P3: [Prog] 3 [EXE]

Here the sample program that has been stored will be executed. The surface (S) and volume (V) for the regular octahedron in the sample problem are computed as:

Length of one side (A)	Surface area (S)	Volume (V)
10 cm	(346.4101615)cm²	(471.4045208)cm³
7	(169.7409791)	(161.6917506)
15	(779.4228634)	(1590.990258)

Operation	Display	
[MODE] 1	_	(Not required if already in the RUN mode.)
[Prog] 1	A?	
10 [EXE] (Value of A)	346.4101615	(S when A=10)
[EXE]	471.4045208	(V when A=10)
[Prog] 1	A?	
7 [EXE] (Value of A)	169.7409791	(S when A=7)
[EXE]	161.6917506	(V when A=7)
[Prog] 1	A?	
15 [EXE] (Value of A)	779.4228634	(S when A=15)
[EXE]	1590.990258	(V when A=15)

*Program computations are performed automatically with each press of [EXE] when it is pressed after data is input or after the result is read.

3-2 PROGRAM CHECKING AND EDITING (CORRECTION, ADDITION, DELETION)

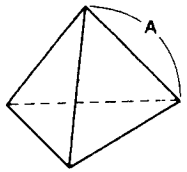
Recalling a stored program can be performed in order to verify its contents. After specifying the desired program area using [⇦] or [⇨] in the WRT mode ([MODE] [2]), the program contents will be displayed by pressing the [EXE] key. Once the program is displayed, the [⇨] (or [⇦]) key is used to advance the program one step at a time for verification.

Pressing [SHIFT] [EXE] causes the cursor to be displayed at the end of the program.

When the program has been improperly stored, editing can also be performed by adding to it or erasing portions. Here a new program will be created by checking and editing the previous sample program (the surface area and volume of a regular octahedron).

EXAMPLE:

Find the surface area and volume of a regular tetrahedron when the length of one side is given.



Length of one side (A)	Surface area (S)	Volume (V)
10 cm	( )cm²	( )cm³
7.5	( )	( )
20	( )	( )

① Formulas

For a surface area S, volume V and one side A, S and V for a regular tetrahedron is defined as:

$$S=\sqrt{3}A^2 \qquad V=\frac{\sqrt{2}}{12}A^3$$

② Programming

As with the previous example, the length of one side is stored in memory A and the program then constructed.

Numeric value A→ [ALPHA] [A] [EXE]

[√] 3 [×] [ALPHA] [A] [x^2] [EXE] ..... S

[√] 2 [÷] 12 [×] [ALPHA] [A] [x^y] 3 [EXE] V

When the above is formed into a program, it appears as follows:

[SHIFT] [?] [→] [ALPHA] [A] [:] [√] 3 [×] [ALPHA] [A] [x^2] [SHIFT] [◢]
[√] 2 [÷] 12 [×] [ALPHA] [A] [x^y] 3

③ Program editing

First, a comparison of the two programs would be helpful.

Octahedron: [SHIFT] [?] [→] [ALPHA] [A] [:] 2 [×] [√] 3 [×] [ALPHA] [A] [x^2] [SHIFT] [◢]
[√] 2 [÷] 3 [×] [ALPHA] [A] [x^y] 3

Tetrahedron: [SHIFT] [?] [→] [ALPHA] [A] [:] [√] 3 [×] [ALPHA] [A] [x^2] [SHIFT] [◢]
[√] 2 [÷] 12 [×] [ALPHA] [A] [x^y] 3

The octahedron program can be changed to a tetrahedron program by deleting the parts marked with wavy lines, and changing those that are marked with straight lines.

In actual practice, this would be performed as follows:

Operation	Display	
[MODE] [2]	WRT 80 Prog _234	
[EXE]	WRT 0 ?→A:2√3A²◢√2÷3	(Displayed from the beginning. To display from the end, press [SHIFT] [EXE] .)
[⇨] [⇨] [⇨] [⇨]	WRT 4 ?→A:2√3A²◢√2÷3	(Move the cursor to the position to be deleted.)
[DEL] [DEL]	WRT 4 ?→A:√3A²◢√2÷3A	(Delete two characters.)
[⇨] ..... [⇨]	WRT 13 ?→A:√3A²◢√2÷3A	(Move the cursor to the position to be corrected.)
[SHIFT] [INS]	WRT 13 ?→A:√3A²◢√2÷3A	(Specify the insert mode.)
[1] [→] [←] [2]	WRT 15 ?→A:√3A²◢√2÷12	(Correct the numeric value.)
[MODE] [1]	_	(After editing is complete, return to the RUN mode.)

④ Program execution

Now this program will be executed.

Length of one side (A)	Surface area (S)	Volume (V)
10 cm	(173.2050808)cm²	(117.8511302)cm³
7.5	(97.42785793)	(49.71844555)
20	(692.820323)	(942.8090416)

Operation	Display
MODE 1	_
Prog 1	A?
10 EXE	173.2050808
EXE	117.8511302
Prog 1	A?
7.5 EXE	97.42785793
EXE	49.71844555
Prog 1	A?
20 EXE	692.820323
EXE	942.8090416

〈Summary〉

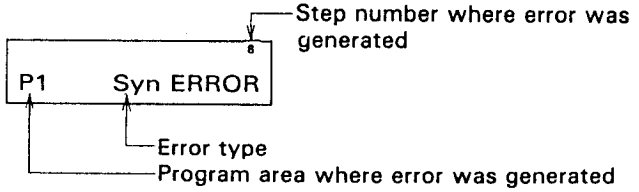
	Operation	Keys used
Program check	•WRT mode specification •Program area specification(Omitted if P1) •Start verification •Verification of contents	MODE 2 ◁ ▷ EXE , SHIFT EXE ◁ ▷
Correction	•Move the cursor to the position to be corrected. •Press correct keys.	◁ ▷
Deletion	•Move the cursor to the position to be deleted. •Delete	◁ ▷ DEL
Insertion	•Move the cursor to the position to be inserted into. •Specify the insert mode. •Press desired keys.	◁ ▷ SHIFT INS

3-3 PROGRAM DEBUGGING (CORRECTING ERRORS)

After a program has been created and input, it will sometimes generate error messages when it is executed, or it will produce unexpected results. This indicates that there is an error somewhere within the program that needs to be corrected. Such programming errors are referred to as "bugs", while correcting them is called "debugging".

■ Debugging when an error message is generated

An error message is displayed as follows:



The error message informs the operator of the program area (P1 to P4) in which the error was generated. It also states the type of error, which gives an idea of the proper countermeasure to be taken. The step number indicates in which step of the program area the error was generated.

■ Error messages

There are a total of four error messages.

① **Syn ERROR** (Syntax error)
Indicates a mistake in the formula or a misuse of program commands.

② **Ma ERROR** (Mathematical error)
Indicates the computation result of a numeric expression exceeds 10^{100}, an illogical operation (i.e. division by zero), or the input of an argument that exceeds the input range of the function.

③ **Go ERROR** (Jump error)
Indicates a missing Lbl for the Goto command.

④ **Stk ERROR** (Stack error)
Indicates the computation performed exceeds the capacity of the stack for numeric values or for commands (see page 15).

Further operation becomes impossible when an error message is displayed. This condition can be released by pressing the [AC] key, but pressing [⇦] or [⇨] before [AC] enters the WRT mode and positions the cursor at the location of the error generated.

■ Checkpoints for each type of error

The following are checkpoints for each type of error:

① **Syn ERROR**
Verify again that there are no errors in the program.
Check to see that there are no mistakes in the arguments used in the program commands (i.e. Goto 10).

② **Ma ERROR**
For computations that require use of the memories, check to see that the numeric values in the memories do not exceed the range of the arguments. This type of error often occurs with division by 0 or the computation of negative square roots.

③ **Go ERROR**
Check to see that there is a corresponding Lbl n when Goto n is used.

④ **Stk ERROR**
Check to see that the formula is not too long thus causing a stack overflow. If this is the case, the formula should be divided into two or more parts.

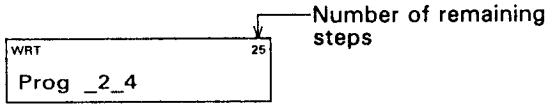
3-4 COUNTING THE NUMBER OF STEPS

The program capacity of this unit consists of a total of 100 steps. The number of steps indicates the amount of storage space available for programs, and it will decrease as programs are input.

The method to determine the current number of remaining steps is as follows.

Specify the WRT mode (MODE 2) and the number of remaining steps will apear at the upper right of the display. At this time the status of the program areas can also be determined.

MODE 2

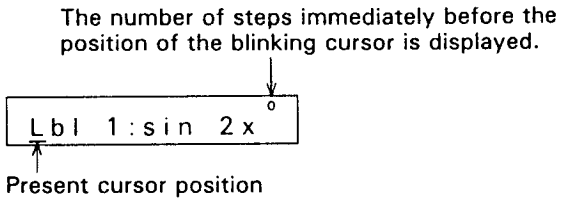


Basically, one function requires a single step, but there are some commands where one function requires two steps.

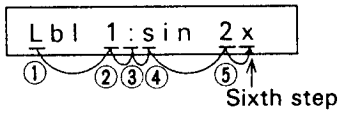
- One function/one step: sin、cos、tan、log、(、)、:、A、B、1、2、3、etc.
- One function/two steps: Lbl 1、Goto 2、etc.

Each step can be verified by the movement of the cursor:

Example:


The number of steps immediately before the position of the blinking cursor is displayed.
0
Lbl 1:sin 2x
Present cursor position

At this time, each press of a cursor key (⇦ or ⇨)will cause the cursor to move to the next sequential step. For example:



3-5 PRECAUTIONS WHEN USING FUNCTIONS

Of the wide variety of functions built into this unit, the following cannot be incorporated into programs:

1. °'" (sexagesimal → decimal), because degrees, minutes, and seconds must each be input.
2. °'" (decimal → sexagesimal), because the displayed value is converted.
3. ENG, ENG (exponent display specification), because the displayed value is converted.

3-6 PROGRAM AREAS AND COMPUTATION MODES

This unit contains a total of 4 program areas (P1 through P4) for the storage of programs. These program areas are all utilized in the same manner, and 4 independent programs can be input. One main program (main routine) and a number of secondary programs (subroutines) can also be stored. The total number of steps available for storage in program areas P1 through P4 is 100 maximum.
Specification of a program area is performed as follows:

RUN mode: Press any key from 1 through 4 after pressing the [Prog] key.

Example: P1 [Prog] [1]

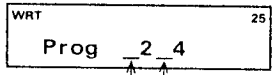
P4 [Prog] [4]

**In this mode, program execution begins when [EXE] is pressed.*

WRT mode: Use [◁] or [▷] to move the cursor under the program area to be specified and press [EXE] .

Only the numbers of the program areas that do not yet contain programs will be displayed. "_" symbols indicate program areas which already contain programs.

Example:



Program areas 1 and 3 already contain programs.

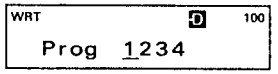
■ Program area and computation mode specification in the WRT mode

Besides normal function computations, to perform binary, octal, decimal and hexadecimal computations and conversions, standard deviation computations, regression computations and complex number computations in a program, a computation mode must be specified. Program mode specification and program area specification are performed at the same time.

First the WRT mode is specified ([MODE] [2]), and then a computation mode is specified. Next, the program area is specified, and, when [EXE] is pressed, the computation mode is memorized in the program area. Henceforth, stored programs will be accompanied with the computation mode.

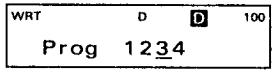
Example: Memorizing the Base-n mode in P2

[MODE] [2]



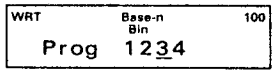
(Assuming that nothing is stored.)

[▷] [▷]



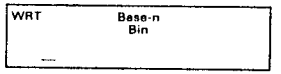
(Specify P2.)

[MODE] [−]



(Specify the Base-n mode.)

[EXE]



As shown above, the computation mode will be memorized into a program area.

**ENG mode can be specified by inputting "Eng" ([MODE] [·]) in a program.*

■ Cautions concerning the computation modes

All key operation available in each computation mode can be stored as programs, but, depending on the computation mode, certain commands or functions cannot be used.

Base-n mode

- Function computations cannot be performed.
- Units of angular measurement cannot be specified.
- All program commands can be used.
- Since a computation result using binary numbers is handled as 32 bits, the whole bits are not displayed at one time.
- The high-order 16 digits can be confirmed using the [Block] key.

SD mode

- Among the functions, Abs and $\sqrt[3]{\ }$ cannot be used.
- Among the program commands, Dsz, > and < cannot be used.

LR mode

- Among the functions, Abs and $\sqrt[3]{\ }$ cannot be used.
- Among the program commands, ⇒, =, ≠, Isz, ≥, ≤, Dsz, > and < cannot be used.

CMPLX mode

- "$\sqrt[3]{\ }$" cannot be used in this mode. (x^2 and x^{-1} can be used in imaginary number part, but other functions can be used only in the real number part or coefficient i for imaginary number part.)

ENG mode

- Function calculations, angle unit specifications, and program commands can all be executed in this mode.

3-7 ERASING PROGRAMS

Erasing programs is performed in the PCL mode. Press [MODE] [3] to specify the PCL mode, and "PCL" will illuminate. There are two methods used to erase programs: erasing a program located in a single program area, and erasing all programs.

■ Erasing a single program

To erase a program in a single program area, specify the PCL mode and press the [AC] key after specifying the program area.

Example: Erase the program in P3 only.

Operation	Display	
[MODE] [3]	PCL 28 Prog _2_4	(Program areas P1 and P3 contain programs.)
[⇨] [⇨]	PCL 28 Prog _2_4	(Move the cursor under P3.)
[AC]	PCL 42 Prog _234	(The program is erased and the number "3" is displayed.)
[MODE] [1]	_	(Return to the RUN mode.)

■ Erasing all programs

To erase all programs stored in program areas 1 through 4, specify the PCL mode and press [SHIFT] and then [DEL].

Example: Delete the programs stored in P1, P2, P3 and P4.

Operation	Display
[MODE] [3]	PCL 11 Prog ____
[SHIFT] [DEL]	PCL 100 Prog 1234

3-8 CONVENIENT PROGRAM COMMANDS

The programs for this unit are made based upon manual computations. Special program commands, however, are available to allow the selection of the formula, and repetitive execution of the same formula.
Here, some of these commands will be used to produce more convenient programs.

■ Jump commands

Jump command are used to change the flow of program execution. Programs are executed in the order that they are input (from the lowest step number first) until the end of the program is reached. This system is not very convenient when there are repeat computations to be performed or when it is desirable to transfer execution to another formula. It is in these cases, however, that the jump commands are very effective. There are three types of jump commands: a simple unconditional jump to a branch destination, a conditional jump that decides the branch destination by whether a certain condition is true or not, and count jump that increases or decreases a specific memory by one and then decides the branch destination after checking whether the value stored equals zero or not.

◆ Unconditional jump

The unconditional jump is composed of "Goto" and "Lbl" . When program execution reaches the statement "Goto n" (where n is a number from 0 through 9), execution then jumps to "Lbl n" (n is the same value as Goto n). The unconditional jump is often used in simple programs to return execution to the beginning for repetitive computations, or to repeat computations from a point within a program.
Unconditional jumps are also used in combination with conditional and count jumps.

Example: The previously presented program to find the surface area and volume of a regular tetrahedron will be rewritten using "Goto 1" and "Lbl 1" to allow repeat computations.

The previous program contained:

?、→、A、:、√、3、×、A、x^2、◢、
√、2、÷、1、2、×、A、x^y、3　　19 steps

**Hereinafter, commas (、) will be used to separate steps for the sake of clarity.*

Add "Goto 1" to the end of the program, and add "Lbl 1" to the beginning of the program as the branch destination.
If this is simply left the way it is, however, the volume will not be displayed and execution will move immediately to the input of one side at the beginning. To prevent this situation, insert a display command (◢) in front of the "Goto 1".

The complete program with the unconditional jump added should look like this:

Lbl、1、:、?、→、A、:、√、3、×、A、x^2、◢、
√、2、÷、1、2、×、A、x^y、3、◢、Goto、1　　25 steps

Now let's try executing this program.

**For details on inputting programs and editing programs, see sections 3-1 and 3-2.*

Operation	Display	
[Prog] [1]	?	(Stored in P1.)
10 [EXE]	173. 2050808	(The length of the side =10)
[EXE]	117. 8511302	
[EXE]	?	
7.5 [EXE]	97. 42785793	(The length of the side =7.5)
[EXE]	49. 71844555	
[EXE]	?	

Since the program is in an endless loop, it will continue execution. To terminate execution, press [MODE] [1] .

[MODE] [1]	_

Besides the beginning of the program, branch destinations can be designated at any point within the program.

Example: Compute $y=ax+b$ when the value for x changes each time, while a and b can also change depending upon the computation.

Program

?、→、A、:、?、→、B、:、Lbl、1、:、?、→、C、:、
A、×、C、+、B、◢、Goto、1　　23 steps

When this program is executed, the values for a and b are stored in memories A and B respectively. After that, only the value for x can be changed.
In this way an unconditional jump is made in accordance with "Goto" and "Lbl", and the flow of program execution is changed. When there is no "Lbl n" to correspond to a "Goto n", an error (Go ERROR) is generated.

◆ Conditional jump

The conditional jump compares a numeric value in memory with a constant or a numeric value in another memory. If the condition is true, the statement following the "⇒" is executed, and if the condition is not true, execution skips the statement and continues following the next ":" or "◢".

Conditional jumps take on the following form:

Left side **Relational operator** **Right side** ⇒**State-ment** { : / ◢ }* **State-ment**

* *Either can be used.*

One memory name (alphabetic character from A through F), constant numeric values or computation formulas (A×2, D−E, etc.) are used for "left side".

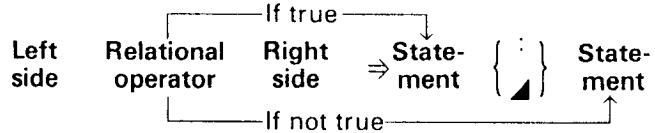
A memory name (A through F) should be used for "right side".

The relational operator is a comparison symbol. There are 6 types of relational operators: =, ≠, ≥, ≤, >, <.

Left side = right side (left side equals right side)
Left side ≠ right side (left side does not equal right side)
Left side ≥ right side (left side is greater than or equal to right side)
Left side ≤ right side (left side is less than or equal to right side)
Left side > right side (left side is greater than right side)
Left side < right side (left side is less than right side)

* = and ≠ are valid in the CMPLX mode, but such comparisons as > and < are only valid when both imaginary numbers are 0 (real number only).

The "⇒" is displayed when SHIFT 7 are pressed. If the condition is true, execution advances to the statement following ⇒. If the condition is not true, the statement following ⇒ is skipped and execution jumps to the statement following the next ":" or "◢".



A statement is a computation formula (sin A×5, etc.) or a program command (Goto, etc.), and everything up to the next ":" or "◢" is regarded as one statement.

Example: If an input numeric value is greater than or equal to zero, compute the square root of that value. If the input value is less than zero, reinput another value.

Program

Lbl、1、:、?、→、A、:、0、≤、A、⇒、√、A、◢、Goto、1

16 steps

In this program, the input numeric value is stored in memory A, and then it is tested to determine whether it is greater than, equal to or less then zero. If the contents of memory A are greater than or equal to 0 (not less than zero), the statement (computation formula) located between "⇒" and "◢" will be executed, and then Goto 1 returns execution to Lbl 1. If the contents of memory A are less than zero, execution will skip the following statement to the next "◢" and returned to Lbl 1 by Goto 1.

Example: Compute the sum of input numeric values. If a 0 is input the total should be displayed.

Program

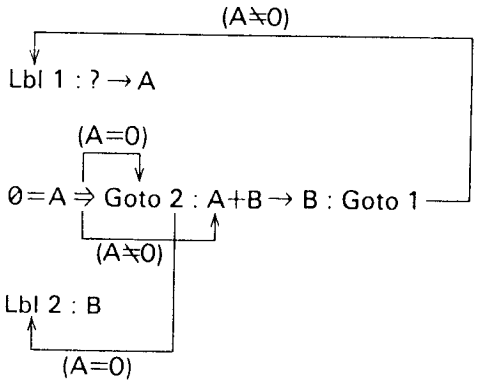
0、→、B、:、
Lbl、1、:、?、→、A、:、0、=、A、⇒、Goto、2、:、
A、+、B、→、B、:、Goto、1、:、
Lbl、2、:、B

31 steps

In this program, a 0 is first stored in memory B to clear it for computation of the sum. Next, the value input by "? → A" is stored in memory A by "0=A⇒" and it is determined whether or not the value stored in memory A equals zero. If 0=A, Goto 2 causes execution to jump to Lbl 2. If memory A does not equal 0, Goto 2 will be skipped and the command A+B→B which follows ":" is executed, and then Goto 1 returns execution to Lbl 1.

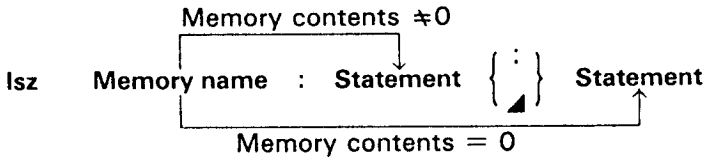
Execution from Lbl 2 will display the sum that has been stored in memory B. Actually, the display command "◢" is inserted following B, but here it can be omitted.

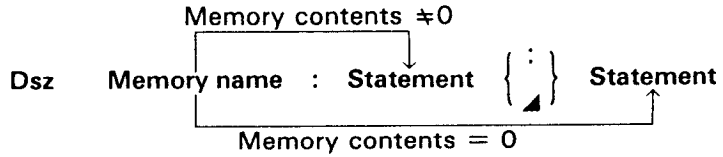
The following illustration shows the flow of the program:



◆ Count jump

The count jump causes the value in a specified memory to be increased or decreased by 1. If the value does equal 0, the following statement is skipped, and the statement following the next ":" or "◢" is executed. The "Isz" command is used to increase the value in memory by 1 and decide the subsequent execution, while the "Dsz" command is used to decrease the value by 1 and decide.





Example: Increase memory A by one ····· Isz A
Decrease memory B by one ····· Dsz B

Example: Determine the average of 10 input numeric values.

Program

1、0、→、A、:、0、→、C、:、
Lbl、1、:、?、→、B、:、B、+、C、→、C、:、
Dsz、A、:、Goto、1、:、C、÷、1、0　　32 steps

In this program, first 10 is stored in memory A, and 0 is stored in memory C. Memory A is used as the "counter" and countdown is performed the specified number of times by the Dsz command. Memory C is used to store the sum of the inputs, and so first must be cleared by inputting a 0.

The numeric value input in response to "?" is stored in memory B, and then the sum of the input values is stored in memory C by "B+C→C".

The statement Dsz A then decreases the value stored in memory A by 1. If the result does not equal 0, the following statement, Goto 1 is executed. If the result equals 0, the following Goto 1 is skipped and "C÷10" is executed.

Example: Determine the altitude at one-second intervals of a ball thrown into the air at an initial velocity of Vm/sec and an angle of S°. The formula is expressed as: $h=V\sin\theta t-\frac{1}{2}gt^2$, with g=9.8, with the effects of air resistance being disregarded.

Program

Deg、:、0、→、C、:、?、→、A、:、?、→、B、:、
Lbl、1、:、Isz、C、:、A、×、sin、B、×、C、−、
9、·、8、×、C、x^2、÷、2、◢、Goto、1
38 steps

In this program the unit of angular measurement is set and memory C is first initialized (cleared). Then the initial velocity and angle are input into memories A and B respectively.

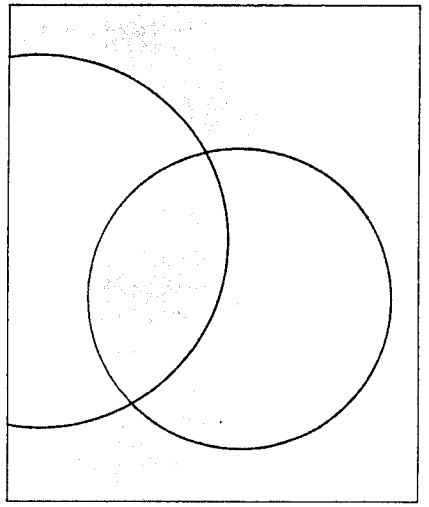
Lbl 1 is used at the beginning of the repeat computations. The numeric value stored in memory C is counted up (increased by 1) by Isz C. In this case、 the Isz command is used only for the purpose of increasing the value stored in memory C, and the subsequent jump does not depend upon any comparison or decision. The Isz command can also be used in the same manner as seen with the Dsz command for jumps that require decisions, but, as can be seen here, it can also be used to simply increase values. If, in place of the Isz command, another method such as "C+1→C" is used, five steps are required instead of the two for the (Isz C) method shown here. Such commands are convenient ways of conserving memory space.

Each time memory C is increased computation is performed according to the formula, and the altitude is displayed. It should be noted that this program is endless, so when the required value is obtained, [MODE] [1] are pressed to terminate the program.

〈**Summary**〉

Command	Formula	Operation
Unconditional jump	Lbl n Goto n (n = natural number from 0 through 9)	Performs unconditional jump to Lbl n corresponding to Goto n
Conditional jump	Left side Relational operator Right side ⇒ State-ment { : ◢ } State-ment (Relational operators: =, ≠, >, <, ≥, ≤)	Left and right sides are compared. If the conditional expression is true, the statement after ⇒ is executed. If not true, execution jumps to the statement following the next : or ◢. Statements include numeric expressions, Goto commands, etc. A memory name (A through F) should be used for "right side".
Count jump	Isz Memory name: State-ment { : ◢ } State-ment Dsz Memory name: State-ment { : ◢ } State-ment (Memory name consists of single character from A through F.)	Numeric value stored in memory is increased (Isz) or decreased (Dsz) by one. If result equals 0, a jump is performed to the statement following the next : or ◢. Statements include numeric expressions, Goto commands, etc.

PROGRAM LIBRARY



CASIO PROGRAM SHEET

Program for: **Greatest common measure** | No. 1

Description

Euclidean general division is used to determine the greatest common measure for two integers a and b.

For $|a|, |b| < 10^9$, positive values are taken as $< 10^{10}$

〈Overview〉

$n_0 = \max(|a|, |b|)$

$n_1 = \min(|a|, |b|)$

$n_k = n_{k-2} - \left[\frac{n_{k-2}}{n_{k-1}}\right] n_{k-1}$

$k = 2, 3 \ldots\ldots$

If $n_k = 0$, then the greatest common measure (c) will be n_{k-1}.

Example

〈1〉	〈2〉	〈3〉
$a = 238$	$a = 23345$	$a = 522952$
$b = 374$	$b = 9135$	$b = 3208137866$
↓	↓	↓
$c = 34$	$c = 1015$	$c = 998$

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode ([MODE] [1])

Step	Key operation	Display	Step	Key operation	Display
1	[Prog] 1	A ?	11		
2	238 [EXE]	B ?	12		
3	374 [EXE]	34.	13		
4	[EXE]	A ?	14		
5	23345 [EXE]	B ?	15		
6	9135 [EXE]	1015.	16		
7	[EXE]	A ?	17		
8	522952 [EXE]	B ?	18		
9	3208137866 [EXE]	998.	19		
10			20		

No. 1

Line	[MODE] [2] Program	Notes	Number of steps
1	Lbl 1 : ? → A : ? → B :		11
2	Abs A → A : Abs B → B :		21
3	B < A ⇒ Goto 2 :		28
4	A → C : B → A : C → B :		40
5	Lbl 2 : (−) (Int (A ÷ B) × B − A		
6	) → C :		59
7	0 = C ⇒ Goto 3 :		66
8	B → A : C → B : Goto 2 :		77
9	Lbl 3 : B ◢ Goto 1		84
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			

Memory contents

A		a, n_0					
B		b, n_1					
C		n_k					
D							
E							
F							

CASIO PROGRAM SHEET

Program for: **Definite integrals using Simpson's rule** | No. 2

Description

$$I=\int_a^b f(x)\,dx=\frac{h}{3}\{y_0+4(y_1+y_3+\cdots+y_{2m-1})+2(y_2+y_4+\cdots+y_{2m-2})+y_{2m}\}$$

$$h=\frac{b-a}{2m}$$

The right-hand portion of the above equation can be transformed as follows.

$$I=\frac{h}{3}\{y_0+\sum_{i=1}^{m}(4y_{2i-1}+2y_{2i})-y_{2m}\}$$

Let $f(x)=\frac{1}{x^2+1}$

Example

〈1〉 $a=0$, $b=1$, $2m=10$

$$I=\int_0^1\frac{1}{x^2+1}dx=0.7853981537$$

〈2〉 $a=2$, $b=5$, $2m=20$

$$I=\int_2^5\frac{1}{x^2+1}dx=0.2662526769$$

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode ([MODE] [1])

Step	Key operation	Display	Step	Key operation	Display
1	[Prog] 1	E ?	11		
2	0 [EXE]	B ?	12		
3	1 [EXE]	F ?	13		
4	10 [EXE]	0.7853981535	14		
5	[Prog] 1	E ?	15		
6	2 [EXE]	B ?	16		
7	5 [EXE]	F ?	17		
8	20 [EXE]	0.2662526768	18		
9			19		
10			20		

No. 2

Line	[MODE] [2] Program															Notes	Number of steps
1	?	→	E	:	?	→	B	:	?	→	F	:					12
2	(	E	x^2	+	1	)	x^{-1}	→	C	:							22
3	(	B	−	E	)	÷	F	→	D	:							32
4	F	÷	2	→	A	:											38
5	Lbl	2	:	E	+	D	→	E	:								47
6	C	+	4	(	E	x^2	+	1	)	x^{-1}	→	C	:				60
7	E	+	D	→	E	:											66
8	C	+	2	(	E	x^2	+	1	)	x^{-1}	→	C	:				79
9	Dsz	A	:	Goto	2	:											85
10	D	(	C	−	(	B	x^2	+	1	)	x^{-1}	)	÷	3			99
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

A	m (Number of repetitions)						
B	b						
C	1						
D	$h=\frac{b-a}{2m}$						
E	a and x						
F	2m						

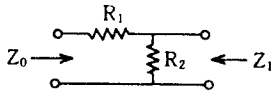
CASIO PROGRAM SHEET

No. 3

Program for: **Minimum loss matching** | No. 3

Description

Calculate R_1 and R_2 which match Z_0 and Z_1 with loss minimized. ($Z_0 > Z_1$)



$$R_1 = Z_0\sqrt{1-\frac{Z_1}{Z_0}} \qquad R_2 = \frac{Z_1}{\sqrt{1-\frac{Z_1}{Z_0}}}$$

Minimum loss $L_{min} = 20 \cdot \log\left(\sqrt{\frac{Z_0}{Z_1}} + \sqrt{\frac{Z_0}{Z_1} - 1}\right)$ [dB]

Example

Calculate the values of R_1, R_2 and L_{min} for $Z_0 = 500\Omega$ and $Z_1 = 200\Omega$.

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode ([MODE] [1])

Step	Key operation	Display	Step	Key operation	Display
1	[Prog] 1	E ?	11		
2	500 [EXE]	F ?	12		
3	200 [EXE]	387.2983346	13		
4	[EXE]	258.1988897	14		
5	[EXE]	8.961393328	15		
6			16		
7			17		
8			18		
9			19		
10			20		

Line	[MODE] [2] Program															Notes	Number of steps
1	?	→	E	:	?	→	F	:									8
2	√	(	1	−	F	÷	E	)	→	A	:						19
3	E	÷	F	→	B	:											25
4	2	Ø	X	log	(	√	B	+	√	(	B	−	1	)	)		
5	→	C	:														43
6	E	×	A	◢													47
7	F	÷	A	◢													51
8	C																52
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

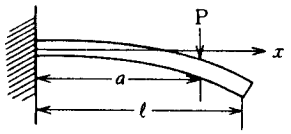
Memory contents

A	$\sqrt{1-\frac{Z_1}{Z_0}}$						
B	$\frac{Z_0}{Z_1}$						
C	L_{min}						
D							
E	Z_0						
F	Z_1						

CASIO PROGRAM SHEET

Program for: **Cantilever under concentrated load** | No. 4

Description



E : Young's modulus [kg/mm²]
I : Geometrical moment of inertia [mm⁴]
a : Distance of concentrated load from support [mm]
P : Load [kg]
x : Distance of point of interest from the support [mm]

Deflection y [mm], Angle of deflection s [°], Bending moment M [kg · mm]

① $l > x > a$

$$y = \frac{Pa^3}{6EI} - \frac{Pa^2}{2EI}x$$

$$s = \tan^{-1}\left(-\frac{Pa^2}{2EI}\right)$$

M = 0 (shearing load Ws = 0)

② $x \leqq a$

$$y = \frac{P}{6EI}x^3 - \frac{Pa}{2EI}x^2$$

$$s = \tan^{-1}\left\{\frac{Px}{2EI}(x-2a)\right\}$$

M = P(x − a) (shearing load Ws = P)

Example

E = 4000kg/mm²
I = 5 mm⁴
a = 30mm
P = 2 kg

What are deflection, angle of deflection, bending moment and shearing load at x = 25 mm and x = 32 mm ?

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode ([MODE] [1])

Step	Key operation	Display	Step	Key operation	Display
1	[Prog] 1	E ?	11	[Prog] 1	E ?
2	4000 [EXE]	F ?	12	E [EXE]	F ?
3	5 [EXE]	A ?	13	F [EXE]	A ?
4	30 [EXE]	C ?	14	A [EXE]	C ?
5	2 [EXE]	D ?	15	C [EXE]	D ?
6	25 [EXE]	−0.6770833333	16	32 [EXE]	−0.66
7	[EXE]	−2.505092867	17	[EXE]	−2.57657183
8	[EXE]	−10.	18	[EXE]	0.
9					
10					

No. 4

Line	[MODE] [2]	Program													Notes	Number of steps
1	? →	E	:	?	→	F	:	?	→	A	:	?	→	C		
2	: ?	→	D	:												20
3	D ≤	A	⇒	Goto	1	:										27
4	C A	x^2	(	A	÷	3	−	D	)	÷	3	E	F	◢		42
5	$\tan^{-1}$	(	(−)	C	A	x^2	÷	2	E	F	◢					53
6	Ø	:	Goto	2	:											58
7	Lbl 1	:	C	D	x^2	(	D	÷	3	−	A	)	÷	2		
8	E F	◢														76
9	$\tan^{-1}$	(	C	D	(	D	−	2	A	)	÷	2	E	F	◢		91
10	C (	D	−	A	:	Lbl	2									99
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																

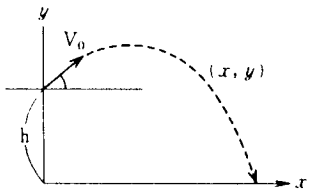
Memory contents

A		a					
B							
C		P					
D		x					
E		E					
F		I					

CASIO PROGRAM SHEET

Program for	No.
Parabolic movement	5

Description



$x = (V_0 \cos a) \cdot t$

$y = (V_0 \sin a) \cdot t - \frac{1}{2} g t^2 + h$

$g = 9.8 [m/s^2]$

V_0 [m/s]

a [°]

Δt [sec.]

h [m]

Example

Initial velocity $V_0 = 130$ (m/sec.)

Initial angle a = 25 (°)

Height h = 0 (m)

$\Delta t = 0.5$ (sec.)

Plot the trace of movement in intervals of Δt.

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode ([MODE] [1])

Step	Key operation	Display	Step	Key operation	Display
1	[Prog] 1	A ?	11	[EXE]	1.
2	25 [EXE]	B ?	12	[EXE]	117.8200123
3	130 [EXE]	C ?	13	[EXE]	50.04037403
4	0.5 [EXE]	D ?	14	[EXE]	1.5
5	0 [EXE]	0.	15	[EXE]	176.7300185
6	[EXE]	0.	16	[EXE]	71.38556104
7	[EXE]	0.	17	[EXE]	2.
8	[EXE]	0.5	18		⋮
9	[EXE]	58.91000616	19		⋮
10	[EXE]	26.24518701	20		

No. 5

Line	[MODE] [2] Program															Notes	Number of steps
1	Deg	:	0	→	F	:											6
2	?	→	A	:	?	→	B	:	?	→	C	:	?	→	D		
3	:																22
4	Lbl	1	:	F	◢									)	)		27
5	B	cos	A	X	F	◢	B	sin	A	X	F	−	9	·	8		
6	F	x^2	÷	2	+	D	→	E	◢								51
7	F	+	C	→	F	:											57
8	0	≤	F	⇒	Goto	1											63
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	

Memory contents

A	a					
B	V_0					
C	Δt					
D	h					
E						
F						

CASIO PROGRAM SHEET

Program for	No.

Description

Example

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1			11		
2			12		
3			13		
4			14		
5			15		
6			16		
7			17		
8			18		
9			19		
10			20		

No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			

Memory contents

A		H		O		V	
B		I		P		W	
C		J		Q		X	
D		K		R		Y	
E		L		S		Z	
F		M		T			
G		N		U			

CASIO PROGRAM SHEET

Program for	No.

Description

Example

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1			11		
2			12		
3			13		
4			14		
5			15		
6			16		
7			17		
8			18		
9			19		
10			20		

No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			

Memory contents

A		H		O		V	
B		I		P		W	
C		J		Q		X	
D		K		R		Y	
E		L		S		Z	
F		M		T			
G		N		U			

CASIO PROGRAM SHEET

Program for	No.

Description

Example

Preparation and operation

- Store the program written on the next page.
- Execute the program as shown below in the RUN mode (MODE 1).

Step	Key operation	Display	Step	Key operation	Display
1			11		
2			12		
3			13		
4			14		
5			15		
6			16		
7			17		
8			18		
9			19		
10			20		

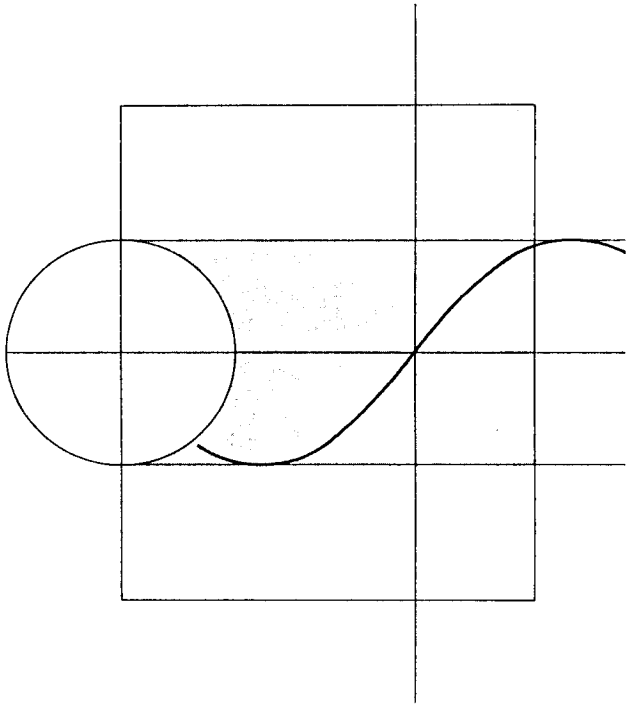
No.

Line	MODE 2 Program	Notes	Number of steps
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			

Memory contents

A		H		O		V	
B		I		P		W	
C		J		Q		X	
D		K		R		Y	
E		L		S		Z	
F		M		T			
G		N		U			

REFERENCE MATERIAL



■ Manual computations

Mode specification	Comp mode (MODE +)	Four arithmetic computations and function computations.
	Base-n mode (MODE −)	Binary, octal, decimal, hexadecimal conversions and computations, logical operations.
	SD mode (MODE ×)	Standard deviation computations (1-variable statistical computations).
	LR mode (MODE ÷)	Regression computations (paired variable statistical computations).
	CMPLX mode (MODE 0)	Complex number computations
	ENG mode (MODE .)	Engineering symbol computations
Functions	Type A functions	Function command input immediately before numeric value. (sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$, log, ln, e^x, 10^x, $\sqrt{}$, $\sqrt[3]{}$, Abs, Int, Frac)
	Type B functions	Function command input immediately after numeric value. (x^2, x^{-1}, $x!$)
	Paired variable functions	Function command input between two numeric values. Numeric value enclosed in parentheses input immediately after function command. (A x^y B (A to the Bth power), B $\sqrt[x]{}$ A (A to the 1/Bth power), A nPr B, A nCr B, Pol(A,B), Rec(A,B)) **A and B are numeric values.*
	Immediately executed functions	Displayed value changed with each press of a key. (ENG, $\overleftarrow{\text{ENG}}$, °′″)

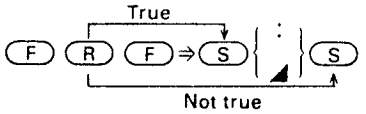
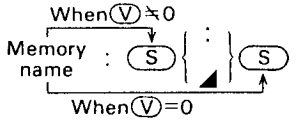
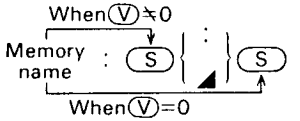
Binary, octal, decimal, hexadecimal computations	Setting number system	Decimal ····· Dec EXE (Dec = $\sqrt{}$) Hexadecimal ····· Hex EXE (Hex = x^2) Binary ····· Bin EXE (Bin = log) Octal ····· Oct EXE (Oct = ln)
	Number system specification	Number system for the numeric value entered immediately after can be specified regardless of the currently set number system. To specify: Decimal ····· SHIFT d (d = $\sqrt{}$) Hexadecimal ····· SHIFT h (h = x^2) Binary ····· SHIFT b (b = log) Octal ····· SHIFT o (o = ln)
	Logical operations	A input numeric value converted to binary and each bit computed. Result converted back to number system used for input, and then displayed. Not .... Reverse of each bit and .... Logical product of each bit or ..... Logical sum of each bit Xor . . . Exclusive logical sum of each bit
Standard deviation computations	Data clear	SHIFT Scl EXE (Scl = AC)
	Data input	Data (;frequency) DT (DT = $\sqrt{}$) **Frequency can be omitted.*
	Data deletion	Data (;frequency) CL (CL = x^2) **Frequency can be omitted.*
	Result display	Number of data (n) ····· ALPHA n EXE (n = 3) Sum (Σx) ····· ALPHA Σx EXE (Σx = 2) Sum of squares (Σx^2) ····· ALPHA Σx^2 EXE (Σx^2 = 1) Mean ($\bar{x}$) ····· SHIFT $\bar{x}$ EXE ($\bar{x}$ = 1) Population standard deviation ($x\sigma_n$) ····· SHIFT $x\sigma_n$ EXE ($x\sigma_n$ = 2) Sample standard deviation ($x\sigma_{n-1}$) ····· SHIFT $x\sigma_{n-1}$ EXE ($x\sigma_{n-1}$ = 3)

Regression computations	Data clear	[SHIFT] [Scl] [EXE] ([Scl] = [AC])
	Data input	x data, y data 〔; frequency〕 [DT] ([DT] = [√]) **Frequency can be omitted.*
	Data deletion	x data, y data 〔; frequency〕 [CL] ([CL] = [x^y]) **Frequency can be omitted.*
	Result display	Number of data (n) ..... [ALPHA] [n] [EXE] ([n] = [3]) Sum of x (Σx)··· [ALPHA] [Σx] [EXE] ([Σx] = [2]) Sum of y (Σy)··· [ALPHA] [Σy] [EXE] ([Σy] = [5]) Sum of squares of x (Σx^2) ..... [ALPHA] [Σx^2] [EXE] ([Σx^2] = [1]) Sum of squares of y (Σy^2) ..... [ALPHA] [Σy^2] [EXE] ([Σy^2] = [4]) Sum of products of x and y (Σxy) ..... [ALPHA] [Σxy] [EXE] ([Σxy] = [6]) Mean of x ($\bar{x}$)··· [SHIFT] [$\bar{x}$] [EXE] ([$\bar{x}$] = [1]) Mean of y ($\bar{y}$)··· [SHIFT] [$\bar{y}$] [EXE] ([$\bar{y}$] = [4]) Population standard deviation of x ($x\sigma_n$) ····· [SHIFT] [$x\sigma_n$] [EXE] ([$x\sigma_n$] = [2]) Population standard deviation of y ($y\sigma_n$) ····· [SHIFT] [$y\sigma_n$] [EXE] ([$y\sigma_n$] = [5]) Sample standard deviation of x ($x\sigma_{n-1}$) ····· [SHIFT] [$x\sigma_{n-1}$] [EXE] ([$x\sigma_{n-1}$] = [3]) Sample standard deviation of y ($y\sigma_{n-1}$) ····· [SHIFT] [$y\sigma_{n-1}$] [EXE] ([$y\sigma_{n-1}$] = [6]) Constant term of regression formula (A) ····· [SHIFT] [A] [EXE] ([A] = [7]) Regression coefficient (B) ..... [SHIFT] [B] [EXE] ([B] = [8]) Correlation coefficient (r) ..... [SHIFT] [r] [EXE] ([r] = [9]) Estimated value of x ($\hat{x}$) ····· y data [SHIFT] [$\hat{x}$] [EXE] ([$\hat{x}$] = [×]) Estimated value of y ($\hat{y}$) ····· x data [SHIFT] [$\hat{y}$] [EXE] ([$\hat{y}$] = [÷])

Special functions	Ans function	The latest result obtained in manual or program computations is stored in memory. It is recalled by pressing [Ans]. **Mantissa of numeric value is 12 digits.*
	Replay function	•After computation results are obtained, the computation formula can be recalled by pressing either [⇦] or [⇨]. •If an error is generated, pressing either [⇦] or [⇨] will cancel the error and the point where the error was generated will be indicated by a blinking cursor.
	Multistatement function	Colons are used to join a series of statements or computation formulas. If joined using "◢", the computation result to that point is displayed.

■ Program computations

Program input	Input mode	WRT mode (MODE 2)
	Computation mode	Mode that conforms with program specified by: MODE +, MODE −, MODE ×, or MODE ÷.
	Program area specification	Cursor is moved to the desired program area number (P1 through P4) using ◁ and ▷, and EXE is pressed.
Program execution	Execution mode	RUN mode (MODE 1)
	Program area specification	Execution starts with Prog program area No.
Program editing	Input mode	WRT mode (MODE 2)
	Program area specification	Cursor is moved to the desired program area number (P1 through P4) using ◁ and ▷, and EXE or SHIFT EXE are pressed.
	Editing	Cursor is moved to position to be edited using ◁ and ▷. •Press correct key for corrections. •Press DEL for deletions. •Press SHIFT INS to specify the insert mode.
Program erasing	Erase mode	PCL mode (MODE 3)
	Erasing a program in a single program area	Cursor is moved to the desired program area number (P1 through P4) using ◁ and ▷, and AC is pressed.
	Erasing the programs in all program areas	Press SHIFT Mcl.

Program commands	Unconditional jump	Program execution jumps to the Lbl *n* which corresponds to Goto *n*. ** n = 1 through 4*
	Conditional jump	If conditional expression is true, the statement after "⇒" is executed. If not true, execution jumps to the statement following next ":" or "◢".  (F): Formula (R): Relational operator (S): Statement ** The relational operator is :* =, ≠, >, <, ≥ or ≤.
	Count jump	The value in a memory is increased or decreased. If the value does not equal 0, the next statement is executed. If it is 0, a jump is performed to the statement following the next ":" or "◢". Increase  Decrease  (S): Statement (V): Value in memory

■ Error messages

Message	Meaning	Countermeasure
Syn ERROR	① Computation formula contains an error. ② Formula in a program contains an error.	① Use ⇦ or ⇨ to display the point where the error was generated and correct it. ② Use ⇦ or ⇨ to display the point where the error was generated and correct the program.
Ma ERROR	① Computation result exceeds computation range. ② Computation is performed outside the input range of a function. ③ Illogical operation (division by zero, etc.)	①②③ Check the input numeric value and correct it. When using memories, check that the numeric values stored in memories are correct.
Go ERROR	• No corresponding Lbl n to Goto n.	• Correctly input a Lbl n to correspond to the Goto n, or delete the Goto n if not required.
Stk ERROR	•Execution of computations that exceed the capacity of the stack for numeric values or stack for computations.	•Simplify the formulas to keep stacks within 6 levels for the numeric values and 14 levels for the computations. •Divide the formula into two or more parts.

■ Input range of functions (general principles)

Function name	Input range
$\sin x$, $\cos x$, $\tan x$	$\|x\| < 9000000000°$ ($50000000\,\pi$ rad, 1×10^{10}grad)
$\sin^{-1}x$, $\cos^{-1}x$	$\|x\| \leqq 1$
$\tan^{-1}x$	$\|x\| < 10^{100}$
e^x	$-10^{100} < x \leqq 230.2585092$
$\sinh x$, $\cosh x$	$x \leqq 230.2585092$
$\tanh x$	$\|x\| < 10^{100}$
$\sinh^{-1}x$	$\|x\| < 5\times10^{99}$
$\cosh^{-1}x$	$1 \leqq x < 5\times10^{99}$
$\tanh^{-1}x$	$\|x\| < 1$
$\log x$, $\ln x$	$1\times10^{-99} \leqq x < 10^{100}$
10^x	$-10^{100} < x < 100$
$\sqrt{x}$	$0 \leqq x < 10^{100}$
x^2	$\|x\| < 10^{50}$
x^{-1} $(1/x)$	$\|x\| < 10^{100}$, $x \neq 0$
$\sqrt[3]{x}$	$\|x\| < 10^{100}$
$x!$	$0 \leqq x \leqq 69$ (x is an integer.)
x^y	$x > 0$: y = real number $x = 0$: $y > 0$ $x < 0$: $y = n,\ {}^1/_2 n + 1$ (n : integer)
$x^{1/y}$	$x > 0$: y = real number $x = 0$: $y > 0$ $x < 0$: $y = 2n + 1,\ {}^1/_n$ ($n \neq 0$) (n : integer)
x^y	When $x<0$, y is a natural number.
$\sqrt[y]{x}$ $(x^{1/y})$	$x \geqq 0$, $y \neq 0$
$n\mathrm{P}r$, $n\mathrm{C}r$	$0 \leqq r \leqq n$, $n < 10^{10}$ (n and r are positive integers or 0.)
Pol (x,y)	$\|x\| < 10^{100}$, $\|y\| < 10^{100}$ However, $\sqrt{x^2+y^2} < 10^{100}$.
Rec (r,θ)	$0 \leqq r < 10^{100}$, $\|\theta\| < 9000000000°$ (50000000π rad, 1×10^{10}grad)

Binary number	(Positive)1111111111111111111111111111111 $\geqq x \geqq 0$ (Negative)1111111111111111111111111111111 $\geqq x \geqq$ 1000000000000000000000000000000
Octal number	(Positive)17777777777 $\geqq x \geqq 0$ (Negative)37777777777 $\geqq x \geqq$ 20000000000
Hexadecimal number	(Positive)7FFFFFFF $\geqq x \geqq 0$ (Negative)FFFFFFFF $\geqq x \geqq$ 80000000
Decimal→ sexagesimal	$\lvert x \rvert \leqq$ 2777777.777. If degrees, minutes and seconds exceed a total of 9 digits, the higher (degrees, minutes) values will be given priority, and displayed in 9 digits.
Statistical computation	$\lvert x \rvert < 10^{50}$, $\lvert y \rvert < 10^{50}$, $\lvert n \rvert < 10^{100}$
Complex calculations (A+Bi and C+Di)	Addition/subtraction $\lvert A \pm C \rvert < 10^{100}$ $\lvert B \pm D \rvert < 10^{100}$ Multiplication $\lvert A\,C \rvert < 10^{100}$ $\lvert B\,D \rvert < 10^{100}$ $\lvert A\,C - B\,D \rvert < 10^{100}$ $\lvert B\,C \rvert < 10^{100}$ $\lvert A\,D \rvert < 10^{100}$ $\lvert B\,C + A\,D \rvert < 10^{100}$ Division $\lvert A\,C \rvert < 10^{100}$ $\lvert B\,D \rvert < 10^{100}$ $\lvert A\,C + B\,D \rvert < 10^{100}$ $\lvert B\,C \rvert < 10^{100}$ $\lvert A\,D \rvert < 10^{100}$ $\lvert B\,C - A\,D \rvert < 10^{100}$

* As a rule, the accuracy of a result is ±1 at the 10th digit.
* Errors may be cumulative with such internal continuous computations with the functions, x^y, $x^{1/y}$, $x!$, $\sqrt[3]{x}$, nPr, nCr and accuracy is sometimes affected.

* In tan$x$, $\lvert x \rvert \neq 90° \times (2n+1)$, $\lvert x \rvert \neq \frac{\pi}{2}\text{rad} \times (2n+1)$, $\lvert x \rvert \neq 100\text{grad}.(2n+1)$ (n is an integer.)
* With sinh x and tanh x, when $x=0$, errors are cumulative and accuracy is affected.
* Certain combinations or permutations may cause errors due to overflow during internal computations.

SPECIFICATIONS

Model: fx-3900P

Computations

Basic computation functions: Negative numbers, exponents, parenthetical addition/subtraction/multiplication/division (with priority sequence judgement function – true algebraic logic).

Built-in functions: Trigonometric/inverse trigonometric functions (units of angular measurement: degrees, radians, grads), hyperbolic/inverse hyperbolic functions, logarithmic/exponential functions, reciprocals, factorials, square roots, cube roots, powers, roots, squares, decimal-sexagesimal conversions, binary-octal-hexadecimal conversions/computations, coordinate transformations, permutations/combinations, π, random numbers, absolute values, integers, fractions, engineering symbol computations, complex number computations,

Statistical computation functions: Standard deviation–number of data, sum, sum of squares, mean, standard deviation (two types)
Linear regression–number of data, sum of x, sum of y, sum of squares of x, sum of squares of y, mean of x, mean of y, standard deviation of x (two types), standard deviation of y (two types), constant term, regression coefficient, correlation coefficient, estimated value of x, estimated value of y

Memories: 6

Computation range: $\pm 1 \times 10^{-99} \sim \pm 9.999999999 \times 10^{99}$ and 0.
Internal operation uses 12-digit mantissa.

Rounding: Performed according to the specified number of significant digits or the number of specified decimal places.

Programs

Number of steps: 100

Jump function: Unconditional jump (Goto), 10 maximum
Conditional jump ($=, \neq, >, <, \geq, \leq$)
Count jump (Isz, Dsz)

Number of stored programs: 4 maximum (P1 to P4)

Check function: Program checking, debugging, deletion, addition, etc.

Common section

Display system and contents: Liquid crystal display (16 digits), 10-digit mantissa and 2-digit exponent, binary, octal, hexadecimal display, sexagesimal display, condition displays (WRT, PCL, Disp, LR, SD, Base-n, CMPLX, D, R, G, S, M, A, P, hyp, Dec, Hex, Bin, Oct, Fix, Sci, Eng)

Error check function: Checks for values exceeding 10^{100}, illogical computations and illogical jumps; error messages displayed.

Power supply: Two lithium batteries (CR2032)

Power consumption: 0.006 W

Battery life: Approximately 590 hours on type CR2032.

Auto power off: Power is automatically switched off approximately 6 minutes after last operation.

Ambient temperature range: 0°C–40°C (32°F–104°F)

Dimensions: 9.2 mmH × 72 mmW × 131 mmD
($\frac{3}{8}$"H × $2\frac{7}{8}$"W × $5\frac{1}{8}$"D)

Weight: 91 g (3.2 oz) including batteries